Command Reference Guide

June 2005

Alcatel. ▪ 26801 West Agoura Rd. Calabasas, CA 91301

818 880 3500

Software Notice

Alcatel assumes no responsibility for product reliability, performance, or both if the user modifies the .CFG file. Full responsibility for any performance issues resulting from modifications made to the .CFG file, by the user, is assumed by the user.

Hardware Notice

The Lithium battery in this product is part of a non-volatile memory device and will retain data for 10 years in the absence of power. Alcatel does not consider the lithium battery in this unit a field replaceable or serviceable part and should not be accessed by the customer.

WARNING: Before working on this equipment be aware of good safety practices and the hazards involved with electrical circuits.

CAUTION: To reduce the risk of fire, use only number 26 AWG or larger UL Listed or CSA Certified Telecommunication Line Cord for all network connections.

CAUTION: Risk of explosion if battery is replaced by an incorrect type. Dispose of used batteries according to the instructions.

Documentation Feedback

The mission of the Technical Publications group at Alcatel is to provide quality documentation that enhances the user's experience with Alcatel products. We are constantly improving our guides and have a genuine interest in ensuring that they are easy to use and enable you to quickly find information you need. We invite you to be part of this process; please email your comments regarding Alcatel product documentation and web content to:

info@ind.alcatel.com

www.alcatel.com
June 2005

CONTENTS

FIGURES

TABLES

ABOUT THIS GUIDE

The *Command Reference Guide* describes the Router command line interface (CLI), providing both instructions and typical command syntax and examples. It complements the *Installation Guide*, which is used to install and maintain your Router system.

This guide is designed for network managers and technicians responsible for the administration of LAN and WAN equipment. Proficiency with networking technologies is assumed.

Organization

The following tables describe the organization and content of this guide.

Table 1 Guide Organization: Major Sections

Section	Description
About This Guide	Defines the user audience, describes the document's organization, introduces special notices, and provides information about other Router user guides.
Command Line Interface	Describes the command line interface (CLI) and how to access navigation and help features. A review of Router configuration standards is included.
Commands	Describes individual CLI commands. Commands are organized by protocol and are in alphabetical order. Each entry provides a command description and syntax and usage examples.

Table 2 Guide Organization: Appendix

Appendix	Section	Description
A	**Alarms and Statistics**	Provides information about system alarms and system statistics.

Notices

Notice paragraphs alert you about issues that require attention. The following paragraphs describe the types of notices used in this guide.

NOTE: A Note offers suggestions for optimal use of your Router system.

ESD: An ESD notice provides information about how to avoid discharge of static electricity and subsequent damage to the Router equipment.

CAUTION: A Caution notice provides information about how to avoid damage to the equipment or application, or to avoid possible service disruption.

WARNING: A Warning notice provides information about how to avoid personal injury.

Documentation

All Router user guides are available in Portable Document Format (PDF). These PDF files are included on the CD-ROM that ships with each Router system. The PDF files are also available on the Router website: **http://eservice.ind.alcatel.com.**

To view PDF files, Adobe Acrobat® Reader® 4.0 must be installed on your PC. If you do not have the Adobe Acrobat Reader installed on your system, you can obtain it free from the Adobe website: **www.adobe.com**.

Other Router Guides

In addition to this guide, Router documentation includes the following:

- *Quick Start Guides*
  A quick start guide is shipped with each system.
- *Installation Guides*
  These guides are designed for network managers and technicians who are responsible for the installation of networking equipment in Telco and service provider network facility environments.
- *Configuration Guides*
  These guides explain how to implement specific features and protocols on Router routers.
- *Router User Guide*
  This user guide explains the general usability features of Router routers.

1

COMMAND LINE INTERFACE

This section introduces the command line interface (CLI) hierarchy and the conventions used to describe it. It also introduces the CLI navigation keys and methods, as well as the available help screens.

Command Types

This guide contains two types of commands: transition, or mode change, commands and standard commands.

Transition commands do not affect the system configuration, they are used to gain access to lower- or next-level commands in the CLI hierarchy. Following each transition command is a brief description, a syntax and usage example, a list of next-level commands, and a list of systems for which the command is applicable.

NOTE: In certain instances, transition commands will select an interface for configuration and access next-level commands. For example, the **configure interface bundle dallas** command access the **configure interface bundle** mode and selects or creates the bundle **dallas**.

Standard commands are used to configure the system. Following each standard command is a brief description, a list of parameters and definitions, a syntax and usage example, a list of related commands, and a list of systems for which the command is applicable.

Context-Sensitive Commands

Some commands are *context-sensitive*. Once a module, bundle, or Ethernet port has been selected for configuration, all further configuration applies only to the selected interface. Table 3 shows a context-sensitive command string for an OmniAccess 604system. In this example, T1 link 1 remains selected for configuration until you exit from the Router model 604/configure/module/t1> prompt.

Table 3 Context-Sensitive Command Sequence

	Context-Sensitive Command String	Example
1	Go into the **configuration** mode.	604>**configure terminal**
2	Specify the type of interface (T1).	604/configure>**module t1**
3	Choose the specific interface (T1 link 1).	604/configure>**module t1 1**
4	From now on, all configuration commands are for T1 link 1 until you exit from module configuration or choose another T1 link.	604/configure/module/t1 1>

NOTE: Command strings that require identification of a specific interface are context-sensitive.

Command Conventions

Each command is briefly described and then followed by the complete syntax, which is essentially a map of the command that shows mandatory and optional parameters. A command example follows, and the systems for which the command applies are listed.

The following tables provide details of the conventions in this guide that are used for syntaxes, examples, and interfaces.

Table 4 Conventions for Syntax

For Syntax	What it means
normal type	Within syntaxes, “normal type” represents required words that must be entered by the user — except when followed by a parameter setting that is enclosed in angled brackets. In that case, only enter the parameter setting enclosed in the angled brackets. **Example 1:** Normal type only. In this example, the user enters the word or argument (module) appearing in the syntax in “normal type.” Syntax: module Command execution: module **Example 2:** Normal type word or argument that is followed by a second normal type word or argument, which is followed by a parameter setting enclosed in angled brackets. In this example, the user enters the first word or argument “connections,” appearing in normal type, and then only enters the value “4” of the second word or argument. Syntax: connections connections < n > Command execution: connections 4 In other words, the first occurence of “connections” must be entered because it is not followed by a setting enclosed in angled brackets. The second occurence of the word “connections” must NOT be entered because it is followed by a setting enclosed in angled brackets. This value of the setting must be entered to execute the command.
[a | b | c]	Normal brackets “[ ]” indicate optional keywords or arguments. A vertical bar “ | “ separates individual settings. **Example:** In this example, the user enters the word “timeout;” must specify either for “tcp” or “udp” for a protocol type; and optionally enters a timeout value “n.” Syntax: timeout protocol_type < tcp | udp > [seconds < n >] Command execution: timeout udp 3600
< >	Angled brackets. All parameter settings are enclosed in angled brackets. The user is directed to choose an appropriate setting. In some cases, the parameter name accompanies the required setting.
[ ]	Optional parameter settings in each syntax are indicated by normal brackets.

Table 5 Conventions for Examples

For Examples	What it means
normal type	Prompts and commands that are part of the main prompt are shown in normal type. **Examples:** 604> 604/display>
bold type	All character strings that a user must enter to execute a command are in bold type. **Example:** 604> **configure term**

Abbreviated Commands

You may enter commands by typing the first few characters of each word in a command string. The Router system recognizes the unique abbreviated entry and executes the command exactly as if you had entered it fully.

For example, to view the currently running system configuration, you may type **display configuration running** at the Router> prompt. You may also type **dis con run** to get the same result. Similarly, you may abbreviate the optional parameter names required by some commands.

For example, a typical entry may be as follows:

mlppp mrru 1600 sequence short seg_threshold 1000 differential_delay 100 discriminator 10.1.100.22

To save time, you may type the following abbreviated string:

mlppp m 1600 seq short seg 1000 diff 100 dis 10.1.100.22

CLI Navigation

The **Tab**, **Esc**, and **Ctrl** keys may be used to move backwards or forwards in the CLI, edit entered command strings, or accelerate the command entry process. Global commands, such as **save**, **ping**, and **display**, may be executed from any level in the CLI hierarchy; they allow the user to execute commonly used commands without exiting their current configuration location.

Navigation Keys

You may use the **Tab** key to quickly enter each word of a command without typing its full name. For example, to enter the **configure** command, you may type its first two letters and then press **Tab** to see the entire word. Then, you may specify an item to configure by pressing the **Spacebar** and then pressing **Tab** repeatedly until the desired sub-command appears. Repeat this sequence for each successive sub-command string until the entire command string appears.

You may also use the other keystrokes shown in Figure 1 during command entry. For example, to back up the cursor without deleting any characters, type **Ctrl-B**. To repeat the last command that you entered, type **Ctrl-P**. To go back several commands, type **Ctrl-P** repeatedly until the desired previous command appears. Or, you may go directly back to the main CLI> prompt from anywhere in the command hierarchy by typing **Ctrl-Z**.

Figure 1 Navigation Keys

```
>  help edit
  key stroke          -- action
  ----------          -- ------
  TAB                 -- command completion
  Esc-B               -- go back one word
  Esc-F               -- forward one word
  Esc-DEL             -- delete one word left to cursor
  BackSpace           -- go back and delete one char
  Ctrl-A              -- start of line
  Ctrl-B / <-         -- go back one char
  Ctrl-D / DEL        -- delete a char
                      -- go up one level if empty command
  Ctrl-E              -- end of line
  Ctrl-F / ->         -- forward one char
  Ctrl-K              -- delete line ahead of cursor
  Ctrl-L              -- refresh line
  Ctrl-N / DN ARROW   -- next command in history
  Ctrl-P / UP ARROW   -- previous command in history
  Ctrl-U              -- delete entire line
  Ctrl-W              -- delete one word left to cursor
>
```

Command Help

Command help is available for navigating the CLI command hierarchy and for assistance with specific commands. You may obtain help by using one of the three commands described below.

Help

Type **help** at the main CLI prompt to see the basic Router system help information. Or, type **help** followed by a command name to view information about that command. Figure 2 shows the help screen.

Figure 2 Help Screen

```
>  help
  ?                    -- display commands under this tree
  exit [level]         -- exit (level nos ) from the current tree
                       --     'exit' from "top level" terminates CLI
  Ctrl-Z               -- exit to top level
  tree                 -- display tree under current node
type 'help edit' to see editing features
type 'help <cmd>' to get help for that command
>
```

Tree

You may view a tree that shows all CLI commands, or a tree that shows only the commands associated with the current command mode (or the routing mode for example). Figure 3 shows two command tree examples. If you type **tree** at the main (604> or equivalent) prompt, the entire list of system commands appears. If you type **tree** within a command mode, such as 604/clear> **tree**, the commands associated with this command mode are displayed.

Figure 3 Router CLI Command Tree

```
 >  tree
 xcli
 |-- ping
 |-- clear
 |    |-- cfg_file
 |    |-- arp
 |    |-- cfg_log
 |    |-- command_log
 |    |-- snmp_stats
 |    |-- counters
 |    |    |-- all
 |    |    |-- ethernet
 |    |    |-- ethernets
 |    |    |-- bundle
 |    |    |-- bundles
 |    |    |-- avc
 |    |    |-- avcs
 |    |    |-- tunnel
 |    |    |-- tunnels
 |    |-- interface
 |    |    |-- all
 |    |    |-- ethernet
 Press any key to continue (q : quit) :
```

? Help Screen

To view help information for a command category, specific command, or a parameter, type the associated word and a question mark (**?**). For example, if you type a question mark at the main command prompt, the system command categories appear. Figure 4 shows a display of these top-level commands.

Figure 4 ? Help Screen

```
>  ?

NAME
  xcli                -- This is root and not a command

SYNTAX
  COMMANDS <cr>

DESCRIPTION
  COMMANDS            -- Any of the following commands can be used

      clear           -- access clear commands
      configure       -- configure from ( flash / network / terminal )
      debug           -- accesses debug commands
      dir             -- directory of files in flash
      erase           -- access erase filesystem commands
      file            -- access file commands
      password        -- Change the user password
      ping            -- invoke ping
      reboot          -- reboot the system
      reload          -- reboot the system
      save            -- save configuration to ( local / network )
      show            -- access show commands
      tclsh           -- To invoke TCL shell
      telnet          -- open a telnet connection
      test            -- access test commands
     trace          -- trace route to destination address or host name
      write           -- write to terminal/network/flash

>
```

NOTE: The default parameters for specific commands appear in parenthesis.

Global Commands

All **display**, **ping**, and **save** commands are available from any level of the CLI. For example, the global **display** commands allow the user to view current configuration settings, alarms, or tests without exiting the **configure** mode. In (Figure 5), a user has displayed a bundle summary while configuring a new bundle.

Similarly, the **ping** and **save** commands are available at any level of the CLI command. The **ping** command verifies connectivity between the Router system and other network hosts; access to the **save** commands from anywhere in the CLI ensures that your configurations may be saved periodically.

Figure 5 Global Display Command

```
>  show configuration
      : Select type of 'configuration' ( Hit Tab )
>  dir

CONTENTS OF /flash1:

  size          date       time       name
--------       ------     ------    --------
 6467513    FEB-04-2004  13:51:22   T1000.1223.Z
 6771268    APR-01-2004  11:38:42   T1000.Z
    1908    APR-01-2004  11:56:18   system.cfg
       0    FEB-05-2004  07:12:30   oldsystem.cfg
 6500329    APR-01-2004  11:49:22   T1000.020404.Z

Total bytes: 19741018
Bytes Free:  12713984
>
```

NOTE: Users can use **show** or **display**. These commands can be used interchangeably.

NOTE: The tab completion feature is not currently available for global commands.

Improved Usability

In this release, the following commands have been modified or added so as to conform with industry standard CLI implementations.

Table 6 Command Changes or Additions

The command was:	The command is now:
(new command)	show ip interface brief
show ip packet_filter filter_list <name>	show ip access-list <name|number>
show ip packet_filter rules (int-name)	show ip access-list-rules <int-name>
show ip packet_filter statistics	show ip access-list-stats <int-name>
show configuration stored	show startup-config
show configuration running	show running-config
show temperature	show environment
show tech	show tech-support
display ...	show ... All display commands are now show commands.
show snmp src_address	show snmp trap-source
show snmp trap_host	show snmp trap-host
show snmp trap_config	show snmp traps
show qos class_templates	This command is has been deleted
clear cfg_file	erase startup-config
new command	clear counters and clear counters all
clear interface bundle	clear counters bundle
clear interface bundles	clear counters bundles
clear interface Ethernet	clear counters Ethernet
clear interface Ethernets	clear counters Ethernet
clear interface avc	clear counters avc
clear interface avcs	clear counters avcs
clear interface tunnel	clear counters tunnel
clear interface tunnels	clear counters tunnels
clear ip packet_filter counters	clear ip access-list counters
file format 0	erase flash
file ls	show flash or dir
file rm filename	erase flash:filename
save local	write memory
configure network	write network
configure flash	configure flash or configure memory
configure network	configure network or write network
Debug fr displayifinfo	Debug fr bundle-info
Debug fr displayvcinfo	Debug fr pvc-info
(new command)	Debug fr mfr states <bundle-name>
(new command)	Debug fr mfr state-machine
(new command)	Debug fr mfr bundle-buffers
(new command)	Debug fr packet inverse-arp
(new command)	Debug fr packet mfr

Table 6 Command Changes or Additions

The command was:	The command is now:
reboot	reload
snmp	snmp-server
snmp system_id	snmp-server chassis-id
snmp enable_trap	snmp-server enable traps
snmp src_address	snmp-server trap-source
snmp trap_host	snmp-server trap-host
conf> ip filter_list	conf> ip access-list
conf> ip apply_filter	conf> ip access-group
conf/Ethernet> ip apply_filter	conf/Ethernet> ip access-group
conf/bundle> ip apply_filter	conf/bundle> ip access-group
conf/bundle/fr/pvc> ip apply_filter	conf/bundle> ip access-group
conf/avc> ip apply_filter	conf/avc> ip access-group
Conf> qos load_class_templates	Deleted
Conf> qos delete_class_templates	Deleted
/configure/Ethernet/qos> add_class class_name parent *[template]* [cr] [br] [priority] [src_ip_address] [dst_ip_address] [netmask] [port] [vlan_id] [dscp] [dot1p] [nat_ip] [mark_dscp] [mark_vlan] [mark_dot1p]	Template parameter has been removed. The command is now: Router/configure/Ethernet/qos> add_class class_name parent [cr] [br] [priority] [src_ip_address] [dst_ip_address] [netmask] [port] [vlan_id] [dscp] [dot1p] [nat_ip] [mark_dscp] [mark_vlan] [mark_dot1p]
Router/configure/bundle/qos> add_class class_name parent *[template]* [cr] [br] [priority] [src_ip_address] [dst_ip_address] [netmask] [port] [vlan_id] [dscp] [dot1p] [nat_ip] [mark_dscp] [mark_vlan] [mark_dot1p]	Template parameter has been removed. The command is now: Router/configure/Ethernet/qos> add_class class_name parent [cr] [br] [priority] [src_ip_address] [dst_ip_address] [netmask] [port] [vlan_id] [dscp] [dot1p] [nat_ip] [mark_dscp] [mark_vlan] [mark_dot1p]
Router/configure/bundle/qos/class> template	Deleted.

.

2 CLEAR

Use the **clear** commands to clear counters, files, logs, statistics, tables, and other data stored by Router systems. The **clear** command clears data for both logical and physical interfaces as well as system features such as IP multiplexing, packet filtering, NAT, QoS, SNMP, and VLAN forwarding.

The first-level **clear** commands are as follows:

Clear Commands
clear arp
erase startup-config
clear cfg_log
clear command_log
clear crypto
clear fr
clear interface
clear ip
clear module
clear qos
clear snmp_stats
clear telnet_session
clear vlanfwd
clear vldfwd
clear vrrp

clear arp

This command clears entries from the address resolution protocol (ARP) table.

Permanent ARP entries are not flushed. After the ARP table is cleared, the system automatically adds new entries as it learns the IP and MAC addresses of connected network hosts.

syntax:

arp

example:

Router/clear> **arp**

The example above clears the ARP table. To view the contents of the ARP table before or after clearing, use the **display arp** command.

related commands:

display arp

applicable models:

All models.

clear cfg_log

This command clears the system configuration log.

The configuration log stores a history of system configuration events; it also records the use of either the **save local** or **save network** commands.

syntax:

cfg_log

example:

Router/clear> **cfg_log**

related commands:

configure network
display configuration running
display configuration stored
save local

applicable models:

All models.

clear command_log

This command clears all information in the command log.

syntax:

command_log

example:

Router /clear> **command_log**

related commands:

display system logging commandLog

applicable models:

All models.

clear counters

This command resets the specified interface counter(s).

syntax:

counters <all | avc | avcs | bundle | bundles | ethernet | ethernets | tunnel | tunnels >

parameter	definition
all	Resets all the interface counters.
avc	Resets the DTE-to-DTE MFR aggregated virtual circuit (AVC) interface.
avcs	Resets all of the DTE-to-DTE MFR aggregated virtual circuit (AVC) interfaces.
bundle	Resets the counters of the specified bundled interface.
bundles	Resets the counters of all of the bundled interfaces.
etherent	Resets the counters for the specified Ethernet interface.
ethernets	Resets the counters for all Ethernet interfaces.
tunnel	Resets the counters for the specified tunnel interface.
tunnels	Resets the counters for all of the tunnel interfaces.

example:

Router/clear> **counters**

applicable models:

All models.

clear counters avc

This command clears all of the counters for every CVC in the AVC and the counters for the AVC itself.

parameter	definition
avc_name	The name of the aggregated virtual circuit (AVC).
dlci	DLCI number of the AVC to be cleared. The range is 16 - 1022.

syntax:

counters avc avc_name < *name* > dlci < n >

example:

Router/clear> **counters avc frame01 100**

related commands:

clear counters avcs
clear counters bundle
clear counters bundles
clear counters Ethernet
clear counters tunnel
clear counters tunnels

applicable models:

All models.

clear counters avcs

This command clears all of the counters for every CVC in all of the AVCs.

syntax:

counters avcs

example:

Router/clear> **counters avcs**

related commands:

clear counters avc
clear counters bundle
clear counters bundles
clear counters Ethernet
clear counters tunnel
clear counters tunnels

applicable models:

All models.

clear counters bundle

This command clears transmission counters on a specific bundle.

parameter	definition
bundle_name	Bundle on which transmission counters will be cleared.

syntax:

counters bundle bundle_name < *name* >

example:

Router/clear> **counters bundle Superior**

The example above clears the transmission counters on the bundle Superior. To view bundle configuration and status before or after clearing counters, use the **show interface bundle** command.

related commands:

clear counters avc
clear counters bundles
clear counters Ethernet
show interface bundle
clear counters tunnel
clear counters tunnels

applicable models:

All models.

clear counters bundles

This command clears the transmission counter on all interface bundles.

syntax:

counters bundles

example:

Router/clear> **counters bundles**

related commands:

clear counters avc
clear counters bundle
clear counters Ethernet
show interface bundles
clear counters tunnel
clear counters tunnels

applicable models:

All models.

clear counters Ethernet

This command clears the transmission counters on an Ethernet port.

parameter	definition
ifnum	Ethernet port on which transmission counters will be cleared (0 or 1).

syntax:

counters ethernet ifnum < 0 | 1 >

example:

Router/clear> **counters ethernet 0**

The example above clears the transmission counters on Ethernet port 0. To view port configuration and status before or after clearing counters, use the **show interface ethernet** command.

related commands:

clear counters avc
clear counters bundle
clear counters bundles
show interface ethernet
clear counters tunnel
clear counters tunnels

applicable models:

All models.

clear counters Ethernets

This command clears the transmission counters on all Ethernet ports.

syntax:

counters ethernets

example:

Router/clear> **counters ethernets**

The example above clears the transmission counters on all Ethernet ports. To view port configuration and status before or after clearing counters, use the **show interface ethernet** command.

related commands:

clear counters avc
clear counters bundle
clear counters bundles
show interface ethernet
clear counters tunnel
clear counters tunnels

applicable models:

All models.

clear counters tunnel

This command clears all of the counters for the specified tunnel.

parameter	definition
tunnel_name	The name of the tunnel (up to eight characters).

syntax:

counters tunnel < *tunnel_name* >

example:

Router/clear> **counters tunnel main**

related commands:

clear counters avcs
clear counters bundle
clear counters bundles
clear counters Ethernet

applicable models:

All models.

clear counters tunnels

This command clears all of the counters for every tunnel.

syntax:

counters tunnels

example:

Router/clear> **counters tunnels**

related commands:

clear counters avcs
clear counters bundle
clear counters bundles
clear counters Ethernet

applicable models:

All models.

clear crypto

This command accesses next-level commands for clearing security-related parameters.

syntax:

crypto

example:

Router/clear> **crypto**

next-level commands

clear crypto ike
clear crypto ipsec
clear crypto statistics

applicable models:

All models.

clear crypto ike

This command accesses next-level commands to clear IKE information.

syntax:

ike

example:

Router/clear/crypto> **ike**

next-level commands

clear crypto ike sa

applicable models:

All models.

clear crypto ike sa

This command clears either all SA entries for a specified policy or all SA entries in the IKE table.

parameter	definition
policy-name	
all	Clears all IKE sa information
policy name	Name of the policy from which the SA entries will be cleared.

syntax:

sa policy-name < all | policy name>

example 1:

To clear all of the SA entries in the IKE table:

Router/clear/> **crypto ike sa all**

example 2:

To clear all of the SA entries for a specified policy:

Router/clear> **crypto ike sa test**

applicable models:

All models.

clear crypto ipsec

This command accesses next-level commands to clear IPSec information.

syntax:

ipsec

example:

Router/clear> **crypto ipsec**

next-level commands

clear crypto ipsec sa

applicable models:

All models.

clear crypto ipsec sa

This command clears either all SA entries for a specified policy or all SA entries in the IPSec table.

parameter	definition
policy-name	
all	Clears all IPSec SA information
policy name	Name of the policy from which the SA entries will be deleted

syntax:

sa policy-name < all | policy name>

example 1:

To clear all of the SA entries in the IPSec table:

Router/clear> **crypto ipsec sa all**

example 2:

To clear all of the SA entries for a specified policy:

Router/clear> **crypto ipsec sa test**

applicable models:

All models.

clear crypto statistics

This command clears all crypto statistics.

syntax:

crypto statistics

example:

To clear all of the crypto statistics, enter:

Router/configure> **clear crypto statistics**

applicable models:

All models.

clear firewall statistics

Clears the firewall statistics.

syntax:Synopsis
clear firewall statistics

example:
To clear the firewall statistics:

Router> clear firewall statistics

clear fr

This command accesses next-level commands for clearing the frame relay inverse ARP table.

syntax:

fr

example:

Router/clear> **fr**

next-level commands

clear fr invarp
clear fr lmistats
clear fr vcstats

applicable models:

All models.

clear fr invarp

This command clears inverse ARP data for all PVCs, PVCs on a specified bundle, or single PVCs.

parameter	definition
all	Clears inverse ARP data for all PVCs.
bundle_name	Name of the bundle to be cleared of inverse ARP data, or name of the bundle to which a specified PVC belongs.
dlci	PVC to be cleared of inverse ARP data The range is 16 - 1022.

syntax:

fr invarp all | bundle_name < *name* > [dlci < n >]

example:

Router/clear> **fr invarp TNET dlci 128**

The example above clears inverse ARP data from PVC 128 on the TNET bundle.

related commands:

clear fr lmistats
clear fr vcstats
display fr invarp
display fr invarp_int

applicable models:

All models.

clear fr lmistats

This command clears lmi statistics from a configured bundle.

syntax:

fr lmistats < bundle_name >

example:

Router/clear> **fr lmistats Boston**

related commands:

clear fr invarp
clear fr vcstats

applicable models:

All models.

clear fr vcstats

This command clears PVC statistics from a configured bundle.

parameter	definition
bundle_name	Configured bundle name
dlci	DLCI number; the range is 16 - 1022.
stat_type	Type of statistics
1	RXMON
2	INJECT
3	1490
4	all (default)

syntax:

vcstats < bundle_name > < dlci > [stat_type]

example:

Router/clear> **fr vcstats Boston 22 3**

related commands:

clear fr invarp
clear fr lmistats

applicable models:

All models.

clear interface

This command accesses next-level commands for clearing counters on bundles and Ethernet ports.

The counters store data such as the number of packets and bytes sent and received, errored packets received, collisions, and up/down states.

syntax:

interface

example:

Router/clear> **interface**

next-level commands

clear interface all
clear counters avc
clear counters bundle
clear counters bundles
clear counters Ethernet

applicable models:

All models.

clear interface all

This command accesses next-level commands for clearing counters on all bundles and all Ethernet ports.

The counters store data such as the number of packets and bytes sent and received, errored packets received, collisions, and up/down states.

syntax:

interface all

example:

Router/clear> **interface all**

next-level commands

clear interface
clear counters bundle
clear counters bundles
clear counters Ethernet

applicable models:

All models.

clear interface avc

This command clears all of the counters for every CVC in the AVC and the counters for the AVC itself.

parameter	definition
avc_name	The name of the aggregated virtual circuit (AVC).
dlci	DLCI number of the AVC to be cleared. The range is 16 - 1022.

syntax:

interface avc avc_name < *name* > dlci < n >

example:

host/clear> **interface avc frame01 100**

related commands:

clear interface avcs
clear interface bundle
clear interface bundles
clear interface ethernet

applicable models:

All models.

clear interface avcs

This command clears all of the counters for every CVC in all of the AVCs.

syntax:

interface avcs

example:

host/clear> **interface avcs**

related commands:

clear interface avc
clear interface bundle
clear interface bundles
clear interface ethernet

applicable models:

All models.

clear interface bundle

This command clears transmission counters on a specific bundle.

parameter	definition
bundle_name	Bundle on which transmission counters will be cleared.

syntax:

interface bundle bundle_name < *name* >

example:

host/clear> **interface bundle Superior**

The example above clears the transmission counters on the bundle Superior. To view bundle configuration and status before or after clearing counters, use the **display interface bundle** command.

related commands:

clear interface avc
clear interface bundles
clear interface ethernet
display interface bundle

applicable models:

All models.

clear interface bundles

This command clears the transmission counter on all interface bundles.

syntax:

interface bundles

example:

host/clear> **interface bundles**

related commands:

clear interface avc
clear interface bundle
clear interface ethernet
display interface bundles

applicable models:

All models.

clear interface ethernet

This command clears the transmission counters on an Ethernet port.

parameter	definition
ifnum	Ethernet port on which transmission counters will be cleared (0 or 1).

syntax:

interface ethernet ifnum < 0 | 1 >

example:

host/clear> **interface ethernet 0**

The example above clears the transmission counters on Ethernet port 0. To view port configuration and status before or after clearing counters, use the **display interface ethernet** command.

related commands:

clear interface avc
clear interface bundle
clear interface bundles
display interface ethernet

applicable models:

All models.

clear interface ethernets

This command clears the transmission counters on all Ethernet ports.

parameter	definition
ifnum	Ethernet port on which transmission counters will be cleared (0 or 1).

example:

host/clear> **interface ethernets**

The example above clears the transmission counters on all Ethernet ports. To view port configuration and status before or after clearing counters, use the **display interface ethernet** command.

related commands:

clear interface avc
clear interface bundle
clear interface bundles
display interface ethernet

applicable models:

All models.

clear interface tunnel

Resets the specified tunnel interface counters to zero.

parameter	definition
tunnel_name	The name of the tunnel.

syntax:

tunnel tunnel_name

example:

host/clear> **interface tunnel main**

related commands:

clear interface tunnels

applicable models:

All models.

clear interface tunnels

Resets the all of the tunnel interfaces counters to zero.

syntax:

clear interface tunnels

example:

host/clear> **interface tunnels**

related commands:

clear interface tunnel

applicable models:

All models.

clear ip

This command accesses next-level commands for clearing network address translation, packet filtering, and routing data from the system.

syntax:

ip

example:

Router/clear> **ip**

next-level commands

clear ip nat
clear ip access-list
clear ip routes

applicable models:

All models.

clear ip access-list

This command accesses next level commands for clearing packet filter data.

syntax:

ip access-list

example:

Router/clear> **ip access-list**

next-level commands

clear ip access-list counters
clear ip access-list statistics

applicable models:

All models.

clear ip access-list counters

This command clears counters for specific or all filtering rule sets.

parameter	definition
name	Name of the filtering rule for which counters are to be cleared.
all	Clears the counters on all filtering rule sets.

syntax:

ip access-list counters < name | all >

example:

Router/clear/ip/access-list> **counters Filter02**

related commands:

clear ip access-list statistics
show ip access-list
show ip access-list filter_list
show ip access-list rules
show ip access-list statistics

applicable models:

All models.

clear ip access-list statistics

This command clears the packet filter statistics for an Ethernet port, bundle, or PVC.

You may also clear the statistics for all interfaces if necessary.

parameter	definition
port number	Ethernet port for which statistics will be cleared (either 0 or 1).
bundle	Bundle for which statistics will be cleared.
bundle:pvc	Bundle and PVC for which statistics will be cleared (frame relay).
all	Clears all filtering statistics.

syntax:

ip access-list statistics < 0 | 1 | bundle | bundle : pvc | all >

example:

Router/clear> **ip access-list statistics wan1:16**

related commands:

clear ip access-list counters
show ip access-list
show ip access-list filter_list
show ip access-list rules
show ip access-list statistics

applicable models:

All models.

clear ip dhcps binding

This command accesses next-level commands for clearing DHCP server binding data from the system.

syntax:

binding

example:

Router/clear> **ip dhcps binding**

related commands:

clear ip dhcps statistics

applicable models:

All models.

clear ip dhcps statistics

This command accesses next-level commands for clearing DHCP server statistics from the system.

syntax:

statistics

example:

Router/clear> **ip dhcps statistics**

related commands:

clear ip dhcps binding

applicable models:

All models.

clear ip nat

This command accesses next-level commands for clearing network address translation data from the system.

syntax:

ip nat

example:

Router/clear/ip> **nat**

next-level commands

clear ip nat all
clear ip nat interface

applicable models:

All models.

clear ip nat all

This command clears NAT translations from all Ethernet ports and bundles on which NAT is enabled.

syntax:

all

example:

Router/clear/ip/nat> **all**

related commands:

clear ip nat interface
show ip nat all

applicable models:

All models.

clear ip nat global

This command accesses next-level commands to clear global NAT configuration.

syntax:

global

example:

Router/clear/ip/nat> **global**

next-level commands

clear ip nat global address
clear ip nat global all
clear ip nat global counters
clear ip nat global dynamic
clear ip nat global port
clear ip nat global static

applicable models:

All models.

clear ip nat global address

This command deletes the specified static IP address from the network address translation (NAT) table.

syntax:

address ip_address

example:

Router/clear/ip/nat/global> **address 10.10.10.5**

applicable models:

All models.

clear ip nat global all

This command deletes all global NAT entries.

syntax:

all

example:

Router/clear/ip/nat/global> **all**

applicable models:

All models.

clear ip nat global counters

This command clears all counters from the network address translation (NAT) table

syntax:

counters

example:

Router/clear/ip/nat/global> **counters**

applicable models:

All models.

clear ip nat global dynamic

This command deletes all dynamic port entries from the network address translation (NAT) table.

syntax:

dynamic

example:

Router/clear/ip/nat/global> **dynamic**

applicable models:

All models.

clear ip nat global port

This command deletes static/dynamic port entries from the network address translation (NAT) table.

parameter	definition
protocol_type	
tcp	Deletes TCP protocol for the specified port.
udp	Deletes UDP protocol for the specified port.
local_address	The local ip address of the port to be deleted.
local_port	The local port to be deleted. The range is 1 - 65535.

syntax:

port protocol_type < tcp | udp > local_address < IP address > local port < n >

example:

Router/clear/ip/nat/global> **port udp 10.10.10.6 100**

applicable models:

All models.

clear ip nat global static

This command deletes all static address/port translation entries.

syntax:

static

example:

Router/clear/ip/nat/global> **static**

applicable models:

All models.

clear ip nat interface

This command accesses next-level commands for clearing NAT data from Ethernet ports and bundles.

syntax:

interface

example:

Router/clear/ip/nat> **interface**

next-level commands

clear ip nat interface bundle
clear ip nat interface ethernet

applicable models:

All models.

clear ip nat interface bundle

This command selects a bundle for clearing NAT data.

After specifying a bundle, use the commands below to clear specific NAT data.

parameter	definition
bundle_name	Name of the bundle to be cleared of NAT data.

syntax:

bundle bundle_name < *name* >

example:

Router/clear/ip/nat/interface> **bundle Austin**

next-level commands

clear ip nat interface bundle address
clear ip nat interface bundle all
clear ip nat interface bundle counters
clear ip nat interface bundle dynamic
clear ip nat interface bundle port
clear ip nat interface bundle static

applicable models:

All models.

clear ip nat interface bundle address

This command clears a static IP address from the bundle translation table.

parameter	definition
ip_address	IP address to be cleared from NAT table.

syntax:

address ip_address< *IP address* >

example:

Router/clear/ip/nat/interface/bundle Austin> **address 10.5.72.1**

related commands:

clear ip nat interface bundle all
clear ip nat interface bundle counters
clear ip nat interface bundle dynamic
clear ip nat interface bundle port
clear ip nat interface bundle static
show ip nat

applicable models:

All models.

clear ip nat interface bundle all

This command clears all NAT data from a bundle translation table.

syntax:

all

example:

Router/clear/ip/nat/interface/bundle Austin> **all**

related commands:

clear ip nat interface address
clear ip nat interface counters
clear ip nat interface dynamic
clear ip nat interface port
clear ip nat interface static
show ip nat

applicable models:

All models.

clear ip nat interface bundle counters

This command clears all NAT event counters on a bundle.

syntax:

counters

example:

Router/clear/ip/nat/interface/bundle Austin> **counters**

related commands:

clear ip nat interface bundle address
clear ip nat interface bundle all
clear ip nat interface bundle dynamic
clear ip nat interface bundle port
clear ip nat interface bundle static

applicable models:

All models.

clear ip nat interface bundle dynamic

This command clears all dynamic IP addresses and port entries from a bundle translation table.

syntax:

dynamic

example:

Router/clear/ip/nat/interface/bundle Austin> **dynamic**

related commands:

clear ip nat interface bundle address
clear ip nat interface bundle all
clear ip nat interface bundle counters
clear ip nat interface bundle port
clear ip nat interface bundle static
show ip nat

applicable models:

All models.

clear ip nat interface bundle port

This command clears application ports from a bundle translation table.

You must specify the type of protocol associated with the port to be cleared (TCP or UDP).

parameter	definition
tcp	TCP protocol
udp	UDP protocol
ip_address	IP address of the port host system
port_number	Number of the port to be cleared from the NAT table. The range is 1 - 65535.

syntax:

port < tcp | udp > ip_address < *IP address* > port_number < n >

example:

Router/clear/ip/nat/interface/bundle Austin> **port tcp 10.4.72.2 305**

related commands:

clear ip nat interface bundle address
clear ip nat interface bundle all
clear ip nat interface bundle counters
clear ip nat interface bundle dynamic
clear ip nat interface bundle static

applicable models:

All models.

clear ip nat interface bundle static

This command clears all static IP addresses and port entries from bundle translation tables.

syntax:

static

example:

Router/clear/ip/nat/interface/bundle Austin> **static**

related commands:

clear ip nat interface bundle address
clear ip nat interface bundle all
clear ip nat interface bundle counters
clear ip nat interface bundle dynamic
clear ip nat interface bundle port
show ip nat

applicable models:

All models.

clear ip nat interface ethernet

This command accesses next-level commands for selecting an Ethernet port for NAT data removal.

After selecting an Ethernet port, use the commands that follow to clear specific NAT data.

parameter	definition
ethernet_identifier	Ethernet port to be cleared of NAT data The values are 0 or 1.

syntax:

ethernet ethernet_identifier < 0 | 1 >

example:

Router/clear/ip/nat/interface> **ethernet 0**

next-level commands

clear ip nat interface ethernet address
clear ip nat interface ethernet all
clear ip nat interface ethernet counters
clear ip nat interface ethernet dynamic
clear ip nat interface ethernet port
clear ip nat interface ethernet static

applicable models:

All models.

clear ip nat interface ethernet address

This command clears a static IP address entry from an Ethernet port translation table.

parameter	definition
ip_address	IP address to be cleared from the NAT table.

syntax:

address ip_address < *IP address* >

example:

Router/clear/ip/nat/interface/ethernet 0> **address 10.1.100.4**

related commands:

clear ip nat interface ethernet all
clear ip nat interface ethernet counters
clear ip nat interface ethernet dynamic
clear ip nat interface ethernet port
clear ip nat interface ethernet static
show ip nat

applicable models:

All models.

clear ip nat interface ethernet all

This command clears all NAT data from an Ethernet port translation table.

syntax:

all

example:

Router/clear/ip/nat/interface/ethernet 0> **all**

related commands:

clear ip nat interface ethernet address
clear ip nat interface ethernet counters
clear ip nat interface ethernet dynamic
clear ip nat interface ethernet port
clear ip nat interface ethernet static
show ip nat

applicable models:

All models.

clear ip nat interface ethernet counters

This command clears all NAT event counters on an Ethernet port.

syntax:

counters

example:

Router/clear/ip/nat/interface/ethernet 0> **counters**

related commands:

clear ip nat interface ethernet address
clear ip nat interface ethernet all
clear ip nat interface ethernet dynamic
clear ip nat interface ethernet port
clear ip nat interface ethernet static
show ip nat

applicable models:

All models.

clear ip nat interface ethernet dynamic

This command clears all dynamic IP address and port entries from a LAN port translation table.

syntax:

dynamic

example:

Router/clear/ip/nat/interface/ethernet 0> **dynamic**

related commands:

clear ip nat interface ethernet address
clear ip nat interface ethernet all
clear ip nat interface ethernet counters
clear ip nat interface ethernet port
clear ip nat interface ethernet static
show ip nat

applicable models:

All models.

clear ip nat interface ethernet port

This command clears application ports (TCP or UDP) from an Ethernet translation table.

parameter	definition
tcp	TCP protocol
udp	UDP protocol
ip address	IP address of the port host system
port number	Number of the port to be cleared from the NAT table. The range is 1 - 65535.

syntax:

port < tcp | udp ? ip address < *IP address* > port number < n >

example:

Router/clear/ip/interface/ethernet 0> **port tcp 10.4.72.2 28**

related commands:

clear ip nat interface ethernet address
clear ip nat interface ethernet all
clear ip nat interface ethernet counters
clear ip nat interface ethernet dynamic
clear ip nat interface ethernet static
show ip nat

applicable models:

All models.

clear ip nat interface ethernet static

This command clears all static IP address and port entries from an Ethernet port translation table.

syntax:

static

example:

Router/clear/ip/nat/interface/ethernet 0> **static**

related commands:

clear ip nat interface ethernet address
clear ip nat interface ethernet all
clear ip nat interface ethernet counters
clear ip nat interface ethernet dynamic
clear ip nat interface ethernet port
show ip nat interface ethernet configuration

applicable models:

All models.

clear ip packet_filter

This command accesses next level commands for clearing packet filter data.

syntax:

ip packet_filter

example:

host/clear> **ip packet_filter**

related commands:

clear ip packet_filter counters
clear ip packet_filter statistics

applicable models:

All models.

clear ip packet_filter counters

This command clears counters for specific or all filtering rule sets.

parameter	definition
name	Name of the filtering rule for which counters are to be cleared.
all	Clears the counters on all filtering rule sets.

syntax:

ip packet_filter counters < name | all >

example:

host/clear/ip/packet_filter> **counters Filter02**

related commands:

clear ip packet_filter statistics
show ip packet_filter
show ip packet_filter filter_list
show ip packet_filter rules
show ip packet_filter statistics

applicable models:

All models.

clear ip packet_filter statistics

This command clears the packet filter statistics for an Ethernet port, bundle, or PVC.

You may also clear the statistics for all interfaces if necessary.

parameter	definition
port number	Ethernet port for which statistics will be cleared (either 0 or 1).
bundle	Bundle for which statistics will be cleared.
bundle:pvc	Bundle and PVC for which statistics will be cleared (frame relay).
all	Clears all filtering statistics.

syntax:

ip packet_filter statistics < 0 | 1 | bundle | bundle : pvc | all >

example 1:

host/clear> **ip packet_filter statistics wan1:16**

related commands:

clear ip packet_filter counters
show ip packet_filter
show ip packet_filter filter_list
show ip packet_filter rules
show ip packet_filter statistics

applicable models:

All models.

clear ipmux

This command accesses next-level commands for clearing IP mux routing data.

syntax:

ipmux

example:

host/clear> **ipmux**

related commands:

clear ipmux routes

applicable models:

All models.

clear ipmux routes

This command clears all static ipmux routes from the system.

syntax:

routes

example:

host/clear/ipmux> **routes**

The example above clear all static ipmux routes from the system. To view the current routes before or after deleting them, use the **display ipmux** command.

related commands:

display ipmux routes
display ip

applicable models:

All models.

clear ip routes

This command clears all dynamic IP routes from the routing table.

syntax:

ip routes

example:

Router/clear> **ip routes**

related commands:

clear ip nat
clear ip access-list
show ip routes
display ipmux routes

applicable models:

All models.

clear ip rtp

This command accesses next-level commands to clear RTP counters and tables.

syntax:

clear ip rtp

example:

Router> **clear ip rtp**

next-level commands

clear ip rtp rxtable
clear ip rtp statistics
clear ip rtp tables
clear ip rtp txtable

applicable models:

All models.

clear ip rtp rxtable

This command clears the RTP receive table for the specified bundle.

syntax:

rxtable interface < bundle name >

example:

Router/clear/ip/rtp> **rxtable wan1**

related commands:

clear ip rtp statistics
clear ip rtp tables
clear ip rtp txtable

applicable models:

All models.

clear ip rtp statistics

This command clears the RTP statistics for the specified bundle.

syntax:

statistics interface < bundle name >

example:

Router/clear/ip/rtp> **statistics wan1**

related commands:

clear ip rtp rxtable
clear ip rtp tables
clear ip rtp txtable

applicable models:

All models.

clear ip rtp tables

This command clears both the RTP receive and transmit tables for the specified bundle for the specified bundle.

syntax:

tables interface <bundle name >

example:

Router/clear/ip/rtp> **tables wan1**

related commands:

clear ip rtp rxtable
clear ip rtp statistics
clear ip rtp txtable

applicable models:

All models.

clear ip rtp txtable

This command clears the RTP transmit table for the specified bundle.

syntax:

txtable interface < bundle name >

example:

Router/clear/ip/rtp> **txtable wan1**

related commands:

clear ip rtp rxtable
clear ip rtp statistics
clear ip rtp tables

applicable models:

All models.

clear ip ssh

This command clears all SSH client sessions or a specific session.

parameter	definition
number	The identification number of a particular SSH session to clear.)

syntax:

ssh [session number <n>]

example 1:

Router1/clear/ip> ssh session 1

example 2:

Router1/clear/ip> ssh

applicable models:

All models.

clear module

This command accesses commands for clearing interface card statistics.

Depending on the system, one or more of the following interfaces types may be cleared: T1 or E1.

syntax:

module

example:

Router/clear> **module**

next-level commands

clear module ct3_userstats
clear module e1_userstats
clear module t1_userstats
clear module t3_userstats
clear module ussi_userstats

applicable models:

All models.

clear module e1_userstats

This command clears statistics on one or more E1 links.

parameter	definition
e1	E1 link(s) for which user statistics will be cleared The range is 1 - 16, depending on the Router system.

syntax:

module e1_userstats e1 < n >

example:

RouterE/clear> **module e1_userstats 4-8**

The example above clears statistics from E1 links 4 through 8. To display statistics before or after clearing them, use the **display module userstats e1** command.

related commands:

clear module ussi_userstats
display module userstats e1

applicable models:

OmniAccess 601, OmniAccess 602

clear module t1_userstats

This command clears statistics on one or more T1 links.

parameter	definition
t1	T1 link(s) for which user statistics will be cleared The range is 1 - 16, depending on the Router system.

syntax:

module t1_userstats t1 < n >

example:

Router/clear> **module t1_userstats 4-8**

The example above clears statistics from T1 links 4 through 8.

To display statistics before or after clearing them, use the **display module userstats t1** command.

related commands:

clear module ct3_userstats
clear module t3_userstats
clear module ussi_userstats
display module userstats t1

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

clear qos

This command accesses next-level commands for clearing QoS statistics.

syntax:

qos

example:

Router/clear> **qos**

next-level commands

clear qos statistics

applicable models:

All models.

clear qos statistics

This command clears the traffic statistics for all classes on a specified bundle, or only on a specific class when that class is specified.

parameter	definition
bundle name	Bundle on which QoS statistics will be cleared for all classes.
class	The class for which QoS stats will be cleared.

syntax:

qos statistics bundle name < *name* > [class < name >]

example:

Router/clear> **qos statistics SrcOne**

The example above clears QoS statistics for all classes of the bundle SrcOne.

related commands:

display qos

applicable models:

All models.

clear snmp_stats

This command clears SNMP statistics.

syntax:

snmp_stats

example:

Router/clear> **snmp_stats**

The example above clears all system SNMP statistics. To view the current SNMP statistics before clearing them, use the **display snmp** commands.

related commands:

display snmp

applicable models:

All models.

clear telnet_session

This command clears an outside Telnet connection.

parameter	definition
seqNo	Telnet session sequence number The range is 1 - 16. Use the **show users** command to display a Telnet session sequence number.

syntax:

telnet_session seqNo < n >

example:

Router/clear> **telnet_session 6**

related commands:

telnet

applicable models:

All models.

clear vlanfwd

This command clears specified VLAN entries.

parameter	definition
management	Clears all entries in the VLAN management table.
statistics	Clears VLAN forwarding statistics. Choose a particular VLAN ID (number) or a range of VLAN IDs The range is 1 - 4095; the default is all.
table	Clears VLAN table. Choose a particular VLAN ID (number) or a range of VLAN IDs The range is 1 - 4095; the default is all.

syntax:

vlanfwd < management | statistics [< n >] | table [< n >] >

example:

Router/clear> **vlanfwd statistics**

related commands:

display vlanfwd

applicable models:

All models.

clear vlanfwd macbridge

This command accesses next-level commands to clear VLAN bridging counters and MAC entries.

syntax:

macbridge

example:

Router> **clear vlanfwd macbridge**

next-level commands

clear vlanfwd macbridge all
clear vlanfwd macbridge dynamic
clear vlanfwd macbridge static
clear vlanfwd macbridge statistics

applicable models:

All models.

clear vlanfwd macbridge all

This command clears all (both dynamic and static) MAC entries in the forwarding database.

syntax:

all

example:

Router> **clear vlanfwd macbridge all**

related commands:

clear vlanfwd macbridge dynamic
clear vlanfwd macbridge static
clear vlanfwd macbridge statistics

applicable models:

All models.

clear vlanfwd macbridge dynamic

This command clears all dynamic MAC entries in the forwarding database.

syntax:

dynamic

example:

Router> **clear vlanfwd macbridge dynamic**

related commands:

clear vlanfwd macbridge all
clear vlanfwd macbridge static
clear vlanfwd macbridge statistics

applicable models:

All models.

clear vlanfwd macbridge static

This command clears all static MAC entries in the forwarding database.

syntax:

static

example:

Router> **clear vlanfwd macbridging static**

related commands:

clear vlanfwd macbridge all
clear vlanfwd macbridge dynamic
clear vlanfwd macbridge statistics

applicable models:

All models.

clear vlanfwd macbridge statistics

This command clears VLAN bridging statistics from all VLAN interfaces.

syntax:

statistics

example:

Router> **clear vlanfwd macbridging statistics**

related commands:

clear vlanfwd macbridge all
clear vlanfwd macbridge dynamic
clear vlanfwd macbridge static

applicable models:

All models.

clear vlanfwd management

This command clears all entries in the vlan management table.

syntax:

management

example:

Router> **clear vlanfwd management**

applicable models:

All models.

clear vlanfwd statistics

This command clears vlan forwarding statistics

parameter	definition
vlan_id	The vlan for which statistics will be cleared. The range is 1 - 4095; the default is all vlans. Enter a single entry (100) or a range (200 - 300).

syntax:

statistics [vlan_id < n | n - n >]

example:

Router> **clear statistics 100**

applicable models:

All models.

clear vlanfwd table

This command clears explicit forwarding entries in the vlan forwarding table.

The implicit entries due to tagging are not cleared with this command. However, adding the vlan_id parameter to the command will clear both the explicit and implicit forwarding table entries due to vlan tagging.

parameter	definition
vlan_id	The vlan for which table entries will be cleared. The range is 1 - 4095; the default is all. Enter a single entry (100) or a range (200 - 300).

syntax:

table [vlan_id < n | n - n >]

example:

Router> **clear vlanfwd table 150-200**

applicable models:

All models.

clear vldfwd

This command accesses next-level vld forwarding clear commands.

syntax:

clear vldfwd

example:

Router> **clear vldfwd**

next-level commands

clear vldfwd statistics
clear vldfwd table

applicable models:

All models.

clear vldfwd statistics

This command clears vld forwarding statistics.

parameter	definition
vld_id	The vld id for which statistics will be cleared. The range is 1 - 4095. Enter a single entry <100> or a range <200-300>; the default is all.

syntax:

statistics [< n | n - n >]

example:

Router> **clear vldfwd statistics 342**

related commands:

clear vldfwd table

applicable models:

All models.

clear vldfwd table

This command clears vld forwarding entries made explicitly by the user using the **add vldid** command.

parameter	definition \
vld_id	The vld id for which statistics will be cleared. The range is 1 - 4095. Enter a single entry <100> or a range <200-300>; the default is all.

syntax:

table [vld_id < n | n - n >]

example:

Router> **clear vldfwd table 125-200**

related commands:

clear vldfwd statistics

applicable models:

All models.

clear vrrp

This command clears vrrp statistics.

parameter	definition
group	The vrrp group for which statistics will be cleared. The range is 1 - 255; the default is all groups.
interface	
ethernet0	Ethernet 0 interface
ethernet1	Ethernet 1 interface

syntax:

vrrp [group group < n >] [interface < ethernet0 | ethernet1 >]

example:

Router> **clear vrrp group 10 ethernet0**

applicable models:

All models.

erase flash

This command erases (formats) the disk.

syntax:

flash [slot-no]

parameter	definition
slot-no	Identifies the slot. The default is 0 for main Flash. Valid range is : 0 - 2.

example:

Router/erase> **flash**

applicable models:

All models.

erase flash-file-name

This command erases the specified file.

syntax:

flash-file-name <file name>

parameter	definition
file name	Specifies the file to be erased.

example:

Router/erase> **erase flash-file-name test.cfg**

applicable models:

All models.

erase startup-config

This command clears the contents of a system configuration file (.CFG).

The file name (but not the configuration) will remain in memory after executing this command.

parameter	definition
file name	Name of file to be cleared. The default is system.cfg.

syntax:

cfg_file file name < *name* >

example:

Router/clear> **cfg_file system.cfg**

The above example clears the contents of the system.cfg file. To verify that you have cleared the configuration file, use the **display configuration stored** or the **show flash** command.

The following screen display shows the contents of system.cfg prior to clearing the file.

screen display example

```
>  dir

CONTENTS OF /flash1:

  size          date       time       name
--------       ------     ------    --------
 6467513    FEB-04-2004  13:51:22   T1000.1223.Z
 6771268    APR-01-2004  11:38:42   T1000.Z
    1908    APR-01-2004  11:56:18   system.cfg
       0    FEB-05-2004  07:12:30   oldsystem.cfg
 6500329    APR-01-2004  11:49:22   T1000.020404.Z

Total bytes: 19741018
Bytes Free:  12713984
>
```

The following screen display shows the contents of system.cfg after clearing the file.

screen display example

```
 >  dir

CONTENTS OF /flash1:

  size          date       time       name
--------       ------     ------    --------
 6467513    FEB-04-2004  13:51:22   T1000.1223.Z
 6771268    APR-01-2004  11:38:42   T1000.Z
       0    APR-01-2004  11:56:18   system.cfg
       0    FEB-05-2004  07:12:30   oldsystem.cfg
 6500329    APR-01-2004  11:49:22   T1000.020404.Z
...
>
```

related commands:

display configuration stored
show flash NCM

applicable models:

All models.

3 CONFIGURE

Use the **configure** commands to configure a Router system. You also can use these commands to subsequently change a system that has been previously configured.

Configuration Methods

Before configuring a system, specify the source from which it will be configured. To do this, go to the main CLI prompt and enter one of the commands described below.

configure flash

Use the **configure flash** command to set the Router system parameters from a configuration file stored in the flash memory. This method is useful if the system configuration differs from the configuration file in flash memory and you want to restore the flash configuration. It is recommended that all configuration files have a .CFG extension.

At the main CLI prompt, type **configure flash** and press **Return**. Then, type the desired configuration file name (with a .cfg extension) and press **Return**, as shown in the example below.

The **configure flash** and **configure memory** commands are equivalents.

syntax:

configure flash

configure memory

example:

Router> **configure flash**

file name (path): **test.cfg**

configure network

Sets the Router system parameters from a configuration file on a network server. This method downloads files via the TFTP protocol, then executes the commands from that file without operator intervention. Use the **configure network** command if you want to configure one or more Router systems with the same configuration.

syntax:

configure network

example:

Router> **configure network**

After you enter the configure network command, press Return, follow the prompt (as shown below), and enter the configuration file path information.

host: **Fremont01**

file name (path): **/networks/system01.cfg**

configure terminal

From a workstation or Telnet session connected to a Router system, **configure terminal** accesses all **configure** commands for setting all system parameters.

syntax:

configure terminal

example:

Router> **configure terminal**

The **configure** commands are as follows:

Configure Commands
configure aaa
configure admin_name
configure arp
configure arp_timeout
configure autoconf
configure boot
configure boot_ic LOCAL
configure boot_ic NCM
configure crypto
configure date
configure echo_errored_cmd
configure event
configure flash
configure fr
configure ftp_server
configure ftp_user
configure header
configure hostname
configure interface
configure ip
configure ipmux
configure ledAnalyzer
configure module
configure network
configure qos
configure reverse_telnet
configure snmp-server
configure sntp
configure SYS_REM
configure SYS_REM_
configure system
configure telnet_banner
configure telnet_timeout
configure terminal
configure user
configure vlanfwd

configure aaa

This command accesses next-level commands to configure parameters for (AAA) Authentication, Authorization, and Accounting.

syntax:

aaa

example:

Router/configure> **aaa**

next-level commands

configure aaa authentication
configure aaa authorization
configure aaa enable
configure aaa radius
configure aaa tacacs

applicable models:

All models.

configure aaa authentication

This command accesses next-level commands to configure lists of authentication methods and protocols.

syntax:

authentication

example:

an-Router/configure/aaa> **authentication**

next-level commands

configure aaa authentication login
configure aaa authentication protocol

applicable models:

All models.

configure aaa authentication login

This command defines a list of authentication methods for login.

parameter	definition
list_name	The name of the login authentication list, either a character string or the word **default**. If list_name is default, all interfaces use this method list without further configuration.
method_list	A list of up to three authentication values, separated by slashes (/), indicating the order in which the methods are used for login on an interface. Possible values are **tacacs**, **radius**, **local**, and **none**.

syntax:

[no] authentication login list_name <name | default> <method_list>

example:

router/configure/aaa> **authentication login default tacacs/radius/local**

related commands:

configure aaa authentication protocol

applicable models:

All models.

configure aaa authentication protocol

This command defines a list of authentication protocols for login.

parameter	definition
list_name	The name of the protocol list, either a character string or the word **default**. If list_name is **default**, all interfaces use this protocol list without further configuration.
protocol_list	A list of up to three protocol values, separated by slashes (/), indicating the order in which the protocols are used for login on an interface. Possible values are **pap**, **chap**, and **ascii**.

syntax:

[no] authentication protocol list_name <name | default> <protocol_list>

example:

Router/configure/aaa> **authentication protocol default pap/ascii**

related commands:

configure aaa authentication login

applicable models:

All models.

configure aaa authorization

This command defines a list of authorization methods for accessing an interface.

parameter	definition
list_name	The name of the authorization list, either a character string or the word **default**. If list_name is default, all interfaces use this method list without further configuration.
method_list	A list of up to two authorization values, separated by slashes (/), indicating the order in which the methods are used for authorization on an interface. Possible values are **tacacs**, **local**, and **none**.

syntax:

[no] authorization commands list_name <name | default> <method_list>

example::

router/configure/aaa> **authorization commands default tacacs/local**

related commands:

configure aaa authentication protocol

applicable models:

All models.

configure aaa enable

This command globally enables the (AAA) Authentication, Authorization, and Accounting mechanism.

parameter	definition
auth	Enables RADIUS authentication
radius	Enables the RADIUS client

syntax:

[no] enable [auth]

example:

Router/configure/aaa> **enable radius**

related commands:

configure aaa radius

applicable models:

All models.

configure aaa radius

This command accesses next-level commands to configure Remote Authentication Dial In User Service (RADIUS) protocol.

syntax:

radius

example:

Router/configure/aaa> **radius**

next-level commands

configure aaa radius auth_port
configure aaa radius fallback
configure aaa radius primary_server
configure aaa radius retries
configure aaa radius secondary_server
configure aaa radius shared_key
configure aaa radius time_out

applicable models:

All models.

configure aaa radius auth_port

This command configures the port used by the RADIUS server for authentication.

parameter	definition
portnumber	The authentication port The default is 1812.

syntax:

[no] auth_port portnumber < n >

example:

Router/configure/aaa/radius> **auth_port 124**

related commands:

configure aaa radius fallback
configure aaa radius primary_server
configure aaa radius retries
configure aaa radius secondary_server
configure aaa radius shared_key
configure aaa radius time_out

applicable models:

All models.

configure aaa radius fallback

This command enables fallback to a local login after three failed login attempts on a console and/or Telnet connection.

parameter	definition
sessions	
all	Configures fallback for all console and Telnet sessions. The default is off.

syntax:

[no] fallback [sessions < all >]

example 1:

This example configures fallback only for the console connection.

Router/configure/aaa/radius> **fallback**

example 2:

This example configures fallback for both a console and a Telnet connection.

Router/configure/aaa/radius> **fallback all**

related commands:

configure aaa radius auth_port
configure aaa radius primary_server
configure aaa radius retries
configure aaa radius secondary_server
configure aaa radius shared_key
configure aaa radius time_out

applicable models:

All models.

configure aaa radius primary_server

This command configures the IP address of the primary RADIUS server.

A primary RADIUS server must be configured to enable RADIUS.

parameter	definition
primary_server ipaddress	The IP address of the primary RAIUS server.

syntax:

primary_server ipaddress

example:

Router/configure/aaa/radius> **primary_server 142.134.14.100**

related commands:

configure aaa radius auth_port
configure aaa radius fallback
configure aaa radius retries
configure aaa radius secondary_server
configure aaa radius shared_key
configure aaa radius time_out

applicable models:

All models.

configure aaa radius retries

This command configures the number of client attempts to communicate with the RADIUS server.

parameter	definition
attempts	The number of attempts to contact the server The range is 1 - 5. The default is 3.

syntax:

retries attempts < n >

example:

Router/configure/aaa/radius> **retries 5**

related commands:

configure aaa radius auth_port
configure aaa radius fallback
configure aaa radius primary_server
configure aaa radius secondary_server
configure aaa radius shared_key
configure aaa radius time_out

applicable models:

All models.

configure aaa radius secondary_server

This command configures the IP address of the secondary RADIUS server.

parameter	definition
secondary_server ipaddress	The IP address of the secondary RAIUS server.

syntax:

secondary_server ipaddress

example:

Router/configure/aaa/radius> **secondary_server 144.120.45.28**

related commands:

configure aaa radius auth_port
configure aaa radius fallback
configure aaa radius primary_server
configure aaa radius retries
configure aaa radius shared_key
configure aaa radius time_out

applicable models:

All models.

configure aaa radius shared_key

This command configures a secret key used by both the RADIUS client and server.

The key is limited to a maximum of 48 characters.

parameter	definition
shared_key key	The value of the shared secret RADIUS key.

syntax:

shared_key key

example:

Router/configure/aaa/radius> **shared_key ax17bfe**

related commands:

configure aaa radius auth_port
configure aaa radius fallback
configure aaa radius primary_server
configure aaa radius retries
configure aaa radius secondary_server
configure aaa radius time_out

applicable models:

All models.

configure aaa radius time_out

This command configures the maximum wait time for a server response.

The maximum time out value is 100.

parameter	definition
time_out	The maximum wait time a RADIUS server waits for a response. Valid range is 1-100 seconds. The default is 8 seconds.

syntax:

time_out seconds < n >

example:

Router/configure/aaa/radius> **time_out 30**

related commands:

configure aaa radius auth_port
configure aaa radius fallback
configure aaa radius primary_server
configure aaa radius retries
configure aaa radius secondary_server
configure aaa radius shared_key

applicable models:

All models.

configure aaa tacacs

This command accesses next-level commands to configure tacacs+ protocol.

syntax:

tacacs

example:

Router/configure/aaa> tacacs

related commands:

configure aaa tacacs primary_server
configure aaa tacacs retries
configure aaa tacacs secondary_server
configure aaa tacacs server_port
configure aaa tacacs shared_key
configure aaa tacacs time_out

applicable models:

All models.

configure aaa tacacs primary_server

This command configures the IP address of the primary tacacs+ server.

A primary tacacs+ server must be configured to enable tacacs+.

parameter	definition
ipaddress	The IP address of the primary tacacs+ server.

syntax:

[no] primary_server ipaddress <address>

example:

Router/configure/aaa/tacacs> primary_server 142.134.14.100

related commands:

configure aaa tacacs retries
configure aaa tacacs secondary_server
configure aaa tacacs server_port
configure aaa tacacs shared_key
configure aaa tacacs time_out

applicable models:

All models.

configure aaa tacacs retries

This command configures the maximum number of client attempts to communicate with the tacacs+ server.

parameter	definition
retries	Specifies the number of retries. The number of attempts to contact the server. The range is 1 - 5. The default is 2.

syntax:

[no] retries retries < n >

example:

Router/configure/aaa/tacacs> retries 5

related commands:

configure aaa tacacs primary_server
configure aaa tacacs secondary_server
configure aaa tacacs server_port
configure aaa tacacs shared_key
configure aaa tacacs time_out

applicable models:

All models.

configure aaa tacacs secondary_server

This command configures the IP address of the secondary tacacs+ server.

parameter	definition
ipaddress	The IP address of the secondary tacacs+ server.

syntax:

[no] secondary_server ipaddress <address>

example:

Router/configure/aaa/tacacs> secondary_server 144.120.45.28

related commands:

configure aaa tacacs primary_server
configure aaa tacacs retries
configure aaa tacacs server_port
configure aaa tacacs shared_key
configure aaa tacacs time_out

applicable models:

All models.

configure aaa tacacs server_port

This command configures the port on the tacacs+ server.

parameter	definition
port	The port on the tacacs+ server. Valid range is 1-65535.

syntax:

server_port port

example:

Router/configure/aaa/tacacs> server_port 6234

related commands:

configure aaa tacacs primary_server
configure aaa tacacs retries
configure aaa tacacs secondary_server
configure aaa tacacs shared_secret
configure aaa tacacs time_out

applicable models:

All models.

configure aaa tacacs shared_key

This command configures a secret key used by both the tacacs+ client and server.

The key is limited to a maximum of eight characters.

parameter	definition
key	The secret key character string.

syntax:

[no] shared_key key <string>

example:

Router/configure/aaa/tacacs> shared_key ax17bfe

related commands:

configure aaa tacacs primary_server
configure aaa tacacs retries
configure aaa tacacs secondary_server
configure aaa tacacs server_port
configure aaa tacacs time_out

applicable models:

All models.

configure aaa tacacs time_out

This command configures the maximum wait time for a tacacs+ server response.

The maximum timeout value is 300.

parameter	definition
seconds	The maximum number of seconds to wait for a response from the tacacs+ server. Valid range is 1-300 seconds.

syntax:

[no] time_out seconds < n >

example:

Router/configure/aaa/tacacs> time_out 30

related commands:

configure aaa tacacs primary_server
configure aaa tacacs retries
configure aaa tacacs secondary_server
configure aaa tacacs server_port
configure aaa tacacs shared_key

applicable models:

All models.

configure admin_name

This command changes the administrator log-in name (Level 1 access) to a user-specified name.

The system default is **Router**.

syntax:

configure admin_name < *new name* >

example:

Router/configure> **admin_name SuperUser**

The example changes the Level 1 user name to SuperUser.

applicable models:

All models.

configure arp

This command adds entries to the ARP table on a Router system.

The ARP table contains the valid IP and MAC addresses of all other network hosts connected to the system. To view the ARP table contents, use the **display arp** command.

parameter	definition
hostip	Host system IP address in the form of aaa.bbb.ccc.ddd
macaddress	Host system MAC address, in six groups of bytes with two digits per byte (nn:nn:nn:nn:nn:nn).
flag	
permanent	Permanent ARP entry, sets permanent flag
published	Sets both the permanent and published flags. The publishing flag lets the system respond to ARP requests even if the system is not the destination host.

syntax:

configure arp < hostip > < macaddress > [flag < published | permanent >]

example:

Router/configure> **arp 10.22.22.22 11:22:33:44:55:66 published**

This example sets both the permanent and published flags.

applicable models:

All models.

configure arp_timeout

This command configures ARP timeout.

parameter	definition
arptimeout	Timeout for arp cache in seconds The range is 60 - 28800; the default is 1200.

syntax:

arp_timeout arptimeout < n >

example:

Router Router/configure> **arp_timeout 1600**

applicable models:

All models.

configure autoconf

This command enables IP mux auto-configuration on the Router system.

Auto-configuration lets one Router system pass Ethernet IP addresses, subnet masks, and source forwarding IP addresses to connected Router systems; this feature removes the need for configuring IP mux on a system-by-system basis.

To avoid problems creating static routes, autoconfig should not be configured "on" when more that 128 bundles are created.

syntax:

[no] autoconf

example:

Router/configure> **autoconf**

The example enables IP mux auto-configuration on the Router system.

applicable models:

All models.

NOTE: IP mux auto-configuration is only applicable when the Router system is operating in IP mux mode. Auto-configurations may be overwritten at connected Router systems.

configure boot

This command configures the boot file from flash.

parameter	definition
file	Boot up file name The default is NCM.Z. The default for the OmniAccess 604 router is T1000.Z.

syntax:

boot < file >

example:

Router/configure> **boot sys.Z**

applicable models:

All models.

configure boot_ic

This command accesses next-level commands to configure the boot file.

syntax:

boot_ic

example:

Router/configure> **boot_ic**

next-level commands

configure boot_ic LOCAL
configure boot_ic NCM

applicable models:

All routers except the 604.

configure boot_ic LOCAL

This command boots the IC with the image from local Flash.

syntax:

LOCAL

example:

Router/configure> **boot_ic LOCAL**

applicable models:

All routers except the 604.

configure boot_ic NCM

This command boots the IC with the image from NCM Flash.

syntax:

NCM

example:

Router/configure> **boot_ic NCM**

applicable models:

All routers except the 604.

configure cabletype

This command selects the type of cable the E1 or monitor port will use.

parameter	definition
e1_port	Specifies the slot number.
coaxial	Chooses coaxial cabling for the E1 port (impedance = 70 ohm). This is the default.
twisted_pair	Chooses twisted pair cabling for the E1 port (impedance = 120 ohm).
monitor_port	Specifies the slot number, either 1 or 2.
coaxial	Chooses coaxial cabling for the monitor port (impedance = 70 ohm).
twisted_pair	Chooses twisted pair cabling for the monitor port (impedance = 120 ohm).

syntax:

cabletype [e1_port < 1 | 2 > < coaxial | twisted_pair >] | [monitor_port < 1 | 2 > < coaxial | twisted_pair >]

example:

RouterE/configure> **cabletype e1_port** 2 **twisted_pair**

applicable models:

OmniAccess 601, OmniAccess 602E

configure date

This command configures the system date.

To view the date and time settings, use the **show date** command. (To set the time, see configure UTC.)

parameter	definition
month	Current month (1-12).
day	Current day of month (1-31).
year	Current year (four digits). Valid range is 1970 - 2100.

syntax:

date month day year

example:

Router/configure> **date 01 07 2005**

In this example, date is January 7, 2005.

related commands:

show date
configure utc

applicable models:

All models.

configure echo_errored_cmd

This command enables or disables reprinting of errored commands.

When enabled, commands that are incorrectly entered are displayed onscreen after pressing the Enter key.

syntax:

[no] echo_errored_cmd

example:

Router/configure> **echo_errored_cmd**

applicable models:

All models.

configure event

This command accesses next-level commands for configuring the event log.

This log stores up to 1000 system events (user logins/logouts, WAN alarms, system failures, etc.). As the log event fills, the Router system clears the oldest entries, one at a time, as new events occur.

To display the event log contents, use the **display event_logs** command. Use the **configure event filter** command to select specific events for display.

syntax:

event

example:

Router/configure> **event**

next-level commands

configure event offline
configure event online

applicable models:

All models.

configure event offline

This command disables the display of system events on your workstation as they occur.

This command does not disable event logging; events are still logged in non-volatile RAM. The default configuration is offline.

syntax:

event offline

example:

Router/configure> **event offline**

related commands:

configure event online

applicable models:

All models.

configure event online

This command enables the real-time display of system events on your workstation.

The default configuration is offline.

syntax:

event online

example:

Router/configure> **event online**

related commands:

configure event offline

applicable models:

All models.

configure firewall nat-failover

Configures the backup interface for the primary interface specified with the nat-ip parameter of firewall policy. Use the **no** form of the command to remove the backup interface for the specified primary interface.

parameter	definition
primary-if	The name of the primary interface.
backup-if	The name of the backup interface for the specified primary interface.

syntax:
firewall nat-failover <primary-if> <backup-if>

example:

To add wan2 as the backup interface for wan1, enter:

```
Router/configure/firewall global> nat-failover wan1 wan2
```

To remove wan2 as the backup interface for wan1, enter:

```
Router/configure/firewall global> no nat-failover wan1 wan2
```

applicable models:

All models.

configure firewall policy

Configures a firewall policy for a specific map. Use the **no** form of the command to delete a policy from the map.There is a maximum of 1024 policies for a map.

parameter	definition
priority	Valid range is from 1 to 1024 which is a unique number for any given map.
address	If not specified, then it is taken as **any**. User is allowed to specify an IP address with a prefix-length or a range of address or a predefined address object for source and destination.
service or the combination of **protocol** and **port** numbers	If not specified, then it is taken as any protocol for any source and destination port numbers.
Source and destination ports	Specified as a single port number or as a range of port numbers.
PAT address	Specified for the **nat-ip** parameter, while defining the firewall policy. Other modes of NAT can be achieved by creating the nat-pool and later attaching the same to the firewall policy
traffic	While configuring firewall policy for a self-traffic, specify **self** for the parameter **traffic**. By default firewall policy is **transit**.

syntax:
firewall policy <priority> <out | in> [permit | deny] [address {src-ip prefix-len dst-ip prefix-len} OR {src-start src-end dst-start dst-end} OR {src-object dst-object}] [service <service-name>] [protocol <tcp | udp | icmp | ah | esp | gre | any | protocol-value>] [port {src-port dst-port} OR {src-start src-end dst-start dst-end}] [traffic <transit | self>] [user-group <name>] [nat-ip <ipaddress | interface>] [port-map] [enable-log| disable-log]

example:

To add a policy to permit ftp from map corp, enter:

```
router/configure> firewall corp
router/configure/firewall corp> policy 1 out service ftp permit
router/configure/firewall corp/policy 1 out>
```

To a policy to permit TCP traffic from (10.1.1.1 with prefix-len 24) to any ip address, enter:

```
router/configure> firewall corp
router/configure/firewall corp> policy 2 out protocol tcp address
10.1.1.1 24 any any permit
router/configure/firewall corp/policy 2 out>
```

To add a policy to permit UDP traffic from (20.1.1.1 - 20.1.1.10) to (30.1.1.1 - 30.1.1.20), enter:

```
router/configure> firewall corp
router/configure/firewall corp> policy 3 out protocol udp address
20.1.1.1 20.1.1.10 30.1.1.1 30.1.1.20 permit
router/configure/firewall corp/policy 3 out>
```

To add a policy to permit Telnet from (40.1.1.1 – 40.1.1.10) to any ip address with NAT (many to o

```
router/configure> firewall corp
router/configure/firewall corp> policy 4 out service telnet address
40.1.1.1 40.1.1.10 any any nat-ip 60.1.1.1 permit
router/configure/firewall corp/policy 4 out>
```

To delete a firewall policy, enter:

```
router/configure> firewall corp
router/configure/firewall corp> no policy 1 out
```

To add a policy to permit FTP from (50.1.1.1 – 50.1.1.10) to any ip address. This example shows how to use the address objects.

```
router/configure> object address addrFtp 50.1.1.1 50.1.1.10
router/configure> firewall corp
router/configure/firewall corp> policy 5 out service ftp address
addrFtp  any permit
router/configure/firewall corp/policy 5 out>
```

To add an inbound policy for web application with destination IP address. Specify the nat-ip address and PAM in the policy command (web is the global default service).

Here the web server is running on port number 8080 in the dmz network. Because of the port-map, all the internet traffic to standard http port 80 will be translated to internal port number 8080.

```
router/configure> firewall dmz
router/configure/firewall dmz> policy 6 in address any 70.1.1.6 32
service web nat-ip 10.1.1.10 nat-port 8080
```

To add an outbound policy for an FTP application from private address (50.1.1.30 – 50.1.1.40) to any IP address. The public address is derived from the outgoing wan interface.

```
router/configure> firewall corp
router/configure/firewall corp> policy 7 out address 50.1.1.30
50.1.1.40 any service ftp nat-ip wan
```

configure flash

This command configures system parameters from a configuration file stored in flash memory.

This method is useful if the system configuration differs from the configuration file in flash memory and you want to restore the flash configuration. It is recommended that all configuration files have a .CFG extension.

At the main CLI prompt, type **configure flash** and press **Return**. Then, type the desired configuration file name with (a .cfg extension) and press **Return**, as shown in the example below.

(The **configure flash** command is equivalent to the **configure memory** command.)

syntax:

flash

memory

example:

Router/configure> **flash**

file name (path): **test.cfg**

related commands:

configure network
configure terminal

applicable models:

All models.

configure fr

This command accesses next-level commands for configuring frame relay operation.

syntax:

fr

example:

Router/configure> **fr**

next-level commands

configure fr invarp
configure fr mfr_e2e_enhanced

applicable models:

All models.

configure fr invarp

This command configures the Inverse ARP polling timer interval.

This is how often the system will poll other frame relay devices for IP data. This function eliminates the need for PVC map entries.

parameter	definition
interval	Time interval for polling The range is 30 - 300 seconds; the default is 30 seconds.

syntax:

fr invarp interval < n >

example:

Router/configure> **fr invarp 60**

related commands:

configure fr mfr_e2e_enhanced

applicable models:

All models.

configure fr mfr_e2e_enhanced

This command switches the system from MFR End-to-End standard FRF.15 mode (default) to MFR End-to-End Router enhanced FRF.15 mode.

Use of this command automatically applies the settings to all existing AVCs without rebooting the system. However, if the settings are saved, they will be used the next time that the system is rebooted.

Enhanced mode calculates differential delay between CVCs; those with unacceptable delay are taken out of active data transfer, thus improving overall throughput of the AVC. Also, end-to-end keepalive messages are sent per CVC, thus helping to maintain the VC integrity across both ends. Keepalive messages also detect software/hardware loopbacks.

syntax:

[no] fr mfr_e2e_enhanced

example:

Router/configure> **fr mfr_e2e_enhanced**

related commands:

configure fr invarp

applicable models:

All models.

configure ftp_server

This command enables and disables the FTP server.

syntax:

[no] ftp_server

example:

Router/configure> **no ftp_server**

related commands:

configure ftp_user
display ftp

applicable models:

All models.

configure ftp_user

This command configures a specific user for logging into an FTP server to transfer files to and from the Router system.

parameter	definition
user name	The name of an approved user. Only one user is supported.

syntax:

configure [no] ftp_user user name < *name* >

example:

Router> **configure ftp_user Alex**

The system will prompt the user to enter a new password.

related commands:

configure ftp_server
display ftp

applicable models:

All models.

configure header

Adds descriptive information to the top of the configuration file. Use the command **save local** after typing in the desired information.

parameter	definition
string	Descriptive information 80 character (maximum) string enclosed in quotation marks

syntax:

header string < "*string*" >

example:

Router/configure> **header "This file was originally activated on 10/17/00."**

applicable models:

All models.

configure hostname

This command changes the Router system host name.

This name becomes the main command prompt. To view the current host name, use the **display hostname** command.

The host name can have up to 15 characters. If you type a name exceeding that length, it will automatically be shortened to **xhost**.

parameter	definition
string	New host name for the system Use up to 15 characters.

syntax:

hostname string < *new host name* >

example:

Router/configure> **hostname Fremont**

applicable models:

All models.

configure interface

This command accesses next-level commands for configuring AVCs, WAN bundles, and Ethernet ports for operation.

Sub-interfaces can be specified.

syntax:

interface

example 1:

Router/configure> **interface**

example 2:

To create sub-interface 1 on Ethernet0, at 192.168.1.20, enter:

Router/configure> **interface ethernet0.1**

Router/configure/ interface ethernet0.1> **ip address 192.168.1.20 255.255.255.0**

next-level commands

configure interface avc
configure interface bundle
configure interface drop_insert
configure interface ethernet
configure interface loopback
configure interface null

applicable models:

All models.

configure interface avc

This command configures a new AVC.

Each AVC requires a unique DLCI number for identification.

parameter	definition
avc_name	Assigns a name to the aggregated virtual circuit.
dlci	DLCI number of the AVC to be configured The range is 16 - 1022.

syntax:

interface avc avc_name < *avc_name* > dlci < n >

example:

Router/configure> **interface avc wan08 22**

next-level commands

configure interface avc bridge
configure interface avc class
configure interface avc cvc
configure interface avc diff_delay
configure interface avc enable
configure interface avc fragment_size
configure interface avc ip
configure interface avc map
configure interface avc red
configure interface avc seg_threshold
configure interface avc sequence
configure interface avc vlan

applicable models:

All models.

configure interface avc bridge

This command configures support for IEEE 802 tagged frames.

parameter	definition
type	
lan	LAN packets (IEEE 802.3) to be bridged This is the default value.
vlan	VLAN packets (IEEE 802.IQ) to be bridged

syntax:

bridge < lan | vlan >

example:

Router/configure/interface/avc wan08 22> **bridge lan**

related commands:

configure interface avc class
configure interface avc cvc
configure interface avc diff_delay
configure interface avc enable
configure interface avc fragment_size
configure interface avc ip
configure interface avc map
configure interface avc red
configure interface avc seg_threshold
configure interface avc sequence
configure interface avc vlan

applicable models:

All models.

configure interface avc class

This command defines the status of the AVC depending on the class selected.

In class D and E, only the administrator can bring the AVC status up or down by modifying the configured shaping parameters of one or more CVCs. In such cases, the remote AVC does not learn of the updated status.

troubleshooting tip:

If the network administrator sets an AVC in the class "D" parameter and then administratively reduces the CIR of any CVC that causes the local AVC to go down (because of the class D setting), the remote AVC will not go down because the local and remote CIR values do not match. To avoid this problem, either change the CIR for the remote CVC to match the local CIR value or reduce the class D CIR value.

parameter	definition
class type	
A	If any CVC is up, the AVC is up. This is the default.
B	If all CVCs are up, the AVC is up.
C	If the user-supplied threshold is satisfied, the AVC is up.
D	If every CVC has a CIR that is greater than or equal to the user-supplied threshold value, the AVC is up.
E	If the total CIR is above the user-supplied threshold value, the AVC is up.

syntax:

class < A | B | C > [< D | E >]

example:

Router/configure/interface/avc wan08 22> **class A**

related commands:

configure interface avc bridge
configure interface avc cvc
configure interface avc diff_delay
configure interface avc enable
configure interface avc fragment_size
configure interface avc ip
configure interface avc map
configure interface avc red
configure interface avc seg_threshold
configure interface avc sequence
configure interface avc vlan

applicable models:

All models.

configure interface avc cvc

This command adds the PVC in a specific bundle, with the DLCI number, to the AVC.

The system can bundle as many as 28 PVCs into one AVC. A CVC cannot be shared between separate AVCs.

parameter	definition
dlci	DLCI of the virtual pvc The range is 16 - 1022.
bundle	Name of the bundle the cvc belongs to

syntax:

cvc dlci < n > bundle < *name* >

example:

Router/configure/interface/avc wan08 22> **cvc 22 wan03**

related commands:

configure interface avc bridge
configure interface avc class
configure interface avc diff_delay
configure interface avc enable
configure interface avc fragment_size
configure interface avc ip
configure interface avc map
configure interface avc red
configure interface avc seg_threshold
configure interface avc sequence
configure interface avc vlan

applicable models:

All models.

configure interface avc diff_delay

This command configures the maximum differential delay allowed for a CVC.

When the AVC is in Router Enhanced FRF.15 mode, using the "no diff_delay" command will disable differential delay calculations and will not take any action toward dropping any CVC that is above the configured differential delay limit. The display of CVCs will, however, display the individual differential delay values for the system Administrator's awareness when the **show interface avc x 16** command is used.

parameter	definition
value	Differential delay in milliseconds. The range is 10 - 128.

syntax:

[no] diff_delay [value < n >]

example:

Router/configure/interface/avc wan08 22> **diff_delay 25**

related commands:

configure interface avc bridge
configure interface avc class
configure interface avc cvc
configure interface avc enable
configure interface avc fragment_size
configure interface avc ip
configure interface avc map
configure interface avc red
configure interface avc seg_threshold
configure interface avc sequence
configure interface avc vlan

applicable models:

All models.

configure interface avc enable avc

This command enables individual AVCs of a frame relay bundle.

syntax:

[no] enable avc

example:
Router/configure/interface/avc wan08 22> **enable avc**

related commands:

configure interface avc enable cvc
configure interface avc enable mfr_e2e_enhanced

applicable models:

All models.

configure interface avc enable cvc

This command enables individual CVCs in this AVC.

parameter	definition
dlci	Data Link Connection Identifier of the CVC Enter a number. The range is 16 - 1022.
bundle_name	The name of the bundle (8-character maximum). Enter a name or string beginning with an alpha character.

syntax:

[no] enable cvc dlci < n > bundle_name < *name* >

example:

Router/configure/interface/avc wan08 22> **enable cvc 30 green2**

related commands:

configure interface avc enable avc
configure interface avc enable mfr_e2e_enhanced

applicable models:

All models.

configure interface avc enable mfr_e2e_enhanced

This command enables Router enhanced FRF.15.

Router Enhanced FRF.15 is a proprietary protocol and can only work between Router systems. In addition to the features of Standard FRF.15, Router Enhanced mode supports end-to-end status integrity of CVCs (doesn't depend on LMI), software loopback detection, and differential delay calculations.

Use the no form of this command to disable enhanced mode and configure standard FRF.15 mode.

syntax:

[no] enable mfr_e2e_enhanced

example:

Router/configure/interface/avc wan08 22> **enable mfr_e2e_enhanced**

related commands:

configure interface avc enable avc
configure interface avc enable cvc

applicable models:

All models.

configure interface avc fragment_size

This command sets the frame size above which a packet is fragmented. Packets are intelligently fragmented when the fragment size is exceeded.

parameter	definition
bytes	Maximum number of bytes in each frame The range is 56 - 4096; the default is 1500.

syntax:

fragment_size bytes < n >

example:

Router/configure/interface/avc wan08 22> **fragment_size 2048**

related commands:

configure interface avc bridge
configure interface avc class
configure interface avc cvc
configure interface avc diff_delay
configure interface avc enable
configure interface avc ip
configure interface avc map
configure interface avc red
configure interface avc seg_threshold
configure interface avc sequence
configure interface avc vlan

applicable models:

All models.

configure interface avc ip access-group

This command applies a filter list to an AVC.

syntax:

[no] ip access-group < *name* >

example:

Router/configure/interface/avc wan08 22> **ip access-group list102000**

next-level commands

configure interface avc ip address
configure interface avc ip directed_broadcast

applicable models:

All models.

configure interface avc ip address

This command assigns a routing destination IP address and subnet to an AVC.

parameter	definition
ip address	IP address of the AVC
subnet mask	Subnet mask of the AVC

syntax:

[no] ip address < *IP address* > subnet mask < *subnet mask* >

example:

Router/configure/interface/avc wan08 22> **ip address 192.5.72.1 255.255.255.0**

next-level commands

configure interface avc ip directed_broadcast
configure interface avc ip access-group

applicable models:

All models.

configure interface avc ip directed_broadcast

This command enables or disables forwarding of direct broadcasts from this interface. The default value for this command is enabled.

syntax:

[no] directed_broadcast

example:

Router/configure/interface/avc wan08 22/ip> **directed_broadcast**

next-level commands

configure interface avc ip address
configure interface avc ip access-group

applicable models:

All models.

configure interface avc ip source_forwarding

This command defines an interface where data on the AVC will be routed to and terminated.

parameter	definition
primary	Source forwarding gateway IP address
secondary	Source forwarding gateway IP address using alternate Ethernet The default is none.

syntax:

[no] source_forwarding primary < *IP address* > [secondary < *IP address* >]

example:

Router/configure/interface/avc wan08 22/ip> **source_forwarding 192.5.72.1**

applicable models:

All models.

configure interface avc map

This command assigns a static route to an AVC.

Once a static route has been assigned, inverse ARP ceases to function. If a destination IP address is changed, the static route will not update.

parameter	definition
ipaddr	IP address of the AVC's destination

syntax:

[no] map ipaddr < *IP address* >

example:

Router/configure/interface/avc wan08 22> **map 10.1.100.20**

related commands:

configure interface avc bridge
configure interface avc class
configure interface avc cvc
configure interface avc diff_delay
configure interface avc enable
configure interface avc fragment_size
configure interface avc ip
configure interface avc red
configure interface avc seg_threshold
configure interface avc sequence
configure interface avc vlan

applicable models:

All models.

configure interface avc red tx_max_thresh

This command sets the maximum threshold for RED.

When the average queue size exceeds this value, RED drops all packets until the average queue size falls below the maximum threshold.

parameter	definition
value	Maximum allowed value of the AVC average queue size. For better results, set the maximum threshold at least twice the value of the minimum threshold setting. The range is 2 - 511; the default is dependent upon the bandwidth of the AVC.

syntax:

tx_max_thresh value < n >

example:

Router/configure/interface/avc wan08 22/red> **tx_max_thresh 14**

related commands:

configure interface avc red tx_min_thresh
configure interface avc red wq_bias_factor

applicable models:

All models.

configure interface avc red tx_min_thresh

This command sets the minimum threshold value for RED on the AVC.

When the average queue size exceeds this threshold, RED starts dropping packets with a probability that increases linearly with the average queue size until the maximum threshold is reached. When the average queue size exceeds the maximum threshold, all packets are dropped.

parameter	definition
value	The threshold at which RED begins dropping packets. Setting the minimum threshold below 10 may cause unnecessary packet drops. The range is 1 - 511; the default is dependent upon the bandwidth of the AVC. The default setting is applied when a bundle is configured. To check the default value, issue the **show interface avc** command.

syntax:

tx_min_thresh value < n >

example:

Router/configure/interface/avc wan08 22/red> **tx_min_thresh 4**

related commands:

configure interface avc red tx_max_thresh
configure interface avc red wq_bias_factor

applicable models:

All models.

configure interface avc red wq_bias_factor

This command sets the weighting factor for the average queue size calculation.

A smaller value causes the average queue to follow the instantaneous value more closely. Whereas, a larger value will cause the average value to change less rapidly and mask the variations in the instantaneous queue size. A very low value of the wq_bias_factor can result in unnecessary packet drops, and a very high value will not respond quickly enough to congestion.

Router recommends that you do not change the **wq_bias_factor** value after setting it.

parameter	definition
value	The weighting factor for the average queue size calculation. The range is 3 - 20; the default is 5.

syntax:

wq_bias_factor wq_bias_factor value < n >

example:

Router/configure/interface/avc wan08 22/red> **wq_bias_factor 6**

related commands:

configure interface avc red tx_min_thresh
configure interface avc red tx_max_thresh

applicable models:

All models.

configure interface avc seg_threshold

This command sets the segmentation threshold for an AVC.
Remember that the segmentation threshold can never be greater than the frame size.

parameter	definition
bytes	All packet fragments will be equal to or greater than seg_threshold. Packets less than 2 x seg_threshold will be forwarded rather than fragmented. The range is 56 - 4096; the default is 512.

syntax:

seg_threshold seg_threshold bytes < n >

example:

Router/configure/interface/avc wan08 22> **seg_threshold 56**

related commands:

configure interface avc bridge
configure interface avc class
configure interface avc cvc
configure interface avc diff_delay
configure interface avc enable
configure interface avc fragment_size
configure interface avc ip
configure interface avc map
configure interface avc red
configure interface avc sequence
configure interface avc vlan

applicable models:

All models.

configure interface avc sequence

This command sets the sequence length of an AVC.
Users can choose either 12- or 24-bit spacing.

parameter	definition
length	
long	24-bit sequence space
short	12-bit sequence space

syntax:

sequence length < long | short >

example:

Router/configure/interface/avc wan08 22> **sequence short**

related commands:

configure interface avc bridge
configure interface avc class
configure interface avc cvc
configure interface avc diff_delay
configure interface avc enable
configure interface avc fragment_size
configure interface avc ip
configure interface avc map
configure interface avc red
configure interface avc seg_threshold
configure interface avc vlan

applicable models:

All models.

configure interface avc vlan

This command accesses next-level commands to configure VLAN tagging on an AVC.

syntax:

vlan

example:

Router/configure/interface/avc wan08 22> **vlan**

next-level commands

configure interface avc vlan router_ip_addr
configure interface avc vlanid

related commands:

configure interface avc bridge
configure interface avc class
configure interface avc cvc
configure interface avc diff_delay
configure interface avc enable
configure interface avc fragment_size
configure interface avc ip
configure interface avc map
configure interface avc red
configure interface avc seg_threshold
configure interface avc sequence

applicable models:

All models.

configure interface avc vlan router_ip_addr

This command configures a remote router IP address for a VLAN ARP request.

syntax:

router_ip_addr ip_address < *IP address* >

example:

Router/configure/interface/avc wan08 22/vlan> **router_ip_addr 100.43.2.1**

related commands:

configure interface avc vlan vlanid

applicable models:

All models.

configure interface avc vlan vlan_ether_type

This command configures the VLAN Ethernet type, specific to this interface.

If this is not configured, the system-wide VLAN Ethertype that was configured using the **configure vlanfwd ether_type** command is used.

parameter	definition
vlan_ether_type	The Ethernet type in decimal format The range is 1 - 65535; the default is 33024 (0x8100).

syntax:

[no] vlan_ether_type vlan_ether_type < n >

example:

Router/configure/interface/avc north 16/vlan> **vlan_ether_type 1500**

applicable models:

All models.

configure interface avc vlan vlanid

This command configures VLAN tagging for an AVC.

parameter	definition
vlanid	VLAN id The range is 1 - 4095.

syntax:

vlanid vlanid < n >

example:

Router/configure/interface/avc wan08 22/vlan> **vlanid 3202**

related commands:

configure interface avc vlan router_ip_addr

applicable models:

All models.

configure interface avc vlan vld_ether_type

This command configures the vld Ethernet type, specific to this interface.

If this is not configured, the system-wide VLAN Ethertype that was configured using the **configure vlanfwd ether_type** command is used.

parameter	definition
vld_ether_type	The Ethernet type in decimal format The range is 1 - 65535; the default is 33024 (0x8100).

syntax:

[no] vld_ether_type vld_ether_type < n >

example:

Router/configure/interface/avc north 16/vlan> **vld_ether_type 1500**

applicable models:

All models.

configure interface avc vlan vldid

This command configures the vld tag for this AVC.

Vld tagging automatically creates a forwarding entry in the vld table. This is an implicit forwarding entry, and is not cleared by the **clear vldfwd table** command.

parameter	definition
vldid	The vld Identification number The range is 4095.

syntax:

[no] vldid vldid < n >

example:

Router/configure/interface/avc north 16/vlan> **vldid 3045**

applicable models:

All models.

configure interface bundle

This command adds and reconfigures existing WAN bundles on all Router systems.

NOTE: The strings “ether0” and “ether1” are key parameters that identify Ethernet interfaces. Do not use these reserve names when naming bundles.

NOTE: The maximum number of all types of bundles cannot exceed 128 per interface.

A bundle consists of one or more physical T1 interfaces bound together as a high-speed (N x 1.544 Mbps) virtual path. For E1, a bundle consists of one or more physical E1 interfaces bound together as a high-speed (N x 2.048 Mbps) virtual path. You also can create fractional T1 or E1 bundles, each consisting of DS0 channels of a T1 or E1 link. A Router system can have up to 128 bundles per interface in any combination of single-T1, multiple-T1, and fractional T1 types.

parameter	definition
bundle_name	Name of the WAN bundle to be configured.

syntax:

configure interface bundle bundle_name < *name* >

example:

Router/configure> **interface bundle Superior**

To see an existing WAN bundle’s configuration before or after changing it, use the **show interface bundle** command. To see a quick summary of all configured bundles currently in the system, use the **show interface bundles** command.

To delete a bundle, type **no interface bundle** followed by that bundle’s name. To confirm bundle removal from the system, use the **show interface bundles** command.

next-level commands

configure interface bundle contact
configure interface bundle description
configure interface bundle drop
configure interface bundle encapsulation
configure interface bundle fr
configure interface bundle hdlc
configure interface bundle icmp
configure interface bundle ip
configure interface bundle ipmux
configure interface bundle link
configure interface bundle mlppp
configure interface bundle nat

configure interface bundle ppp

configure interface bundle qos

configure interface bundle red

configure interface bundle restore

configure interface bundle shutdown

configure interface bundle vlanid

applicable models:

All models.

configure interface bundle bcp

This command accesses next-level commands for configuring a PPP bundle with the Bridging Control Protocol (BCP).

syntax:

bcp

example:

Router/configure/interface/bundle SF_01> **bcp**

next-level commands

configure interface bundle bcp bridge

applicable models:

All models.

NOTE: You must delete the bundle IP address before executing this command.

configure interface bundle bcp bridge

This command configures a PPP bundle with the Bridge Control Protocol.

In the **lan** setting, the system will receive untagged packets, tag them, and forward them; in the **vlan** setting, the system will receive VLAN-tagged packets and automatically forward them. This follows IEEE 802 for tagged frames.

parameter	definition
type	
lan	LAN packets to be bridged (default).
vlan	VLAN packets to be bridged.

syntax:

bridge type < lan | vlan >

example:

Router/configure/interface/bundle SF_01/bcp> **bridge vlan**

applicable models:

All models.

NOTE: When bridging is turned on, IP addresses are not operational on bundles.

configure interface bundle contact

This command specifies a person to be contacted regarding the bundle.

This command entry must be enclosed in quotation marks.

parameter	definition
contact	Name of the person to be contacted Use up to 15 characters; enclosed in quotation marks.

syntax:

contact < *"desired contact information"* >

example:

Router/configure/interface/bundle SF_01> **contact "JohnB x1045"**

applicable models:

All models.

configure interface bundle crypto

This command configures the network type for the specified non-Frame Relay bundle.

There is no default for this command.

parameter	definition
network_type	
trusted	Interface is part of a trusted network
untrusted	Interface is part of a untrusted network
dmz	Interface is part of a dmz network

syntax:

[no] crypto network_type < trusted | untrusted | dmz >

example:

Router1/configure/interface/bundle Dallas> **crypto trusted**

applicable models:

OmniAccess 604

configure interface bundle description

This command gives a brief description of the bundle.

This entry is also optional and must be enclosed in quotation marks.

parameter	definition
descr	A description of the bundle Use up to 15 characters; enclose in quotation marks.

syntax:

description descr <" *description*">

example:

Router/configure/interface/bundle SF_01> **description "Fremont"**

applicable models:

All models.

configure interface bundle drop

This command allows the Router system to drop an errored T1 or E1 link from the bundle.

A bundle is dropped if excessive errors of a user-specified type occur for a user-defined time interval.

parameter	definition
error_type	Error condition that will cause droppage of a T1 or E1 link from the bundle, as follows:
ses	Severely Errored Seconds
es	Errored Seconds
uas	Unavailable Seconds
eev	Excessive Error Violations Seconds
bes	Bursty Errored Seconds
bpv	Bipolar Violations
lofc	Loss of Frame Counts
css	Controlled Slip Seconds
oof	Out of Frame Seconds
crc	CRC-6 Errors
drop_time	Drop time in consecutive seconds (es, ses, uas) or drop counts.

syntax:

drop error_type < ses | es | uas | eev | bes | bpv | lofc | css | oof | crc > drop_time < n >

example:

Router/configure/interface/bundle SF_01> **drop ses 15**

In this example, a T1 link will be dropped from the bundle if more than 15 consecutive SES occur.

applicable models:

All models.

NOTE: This parameter does not apply for single-T1 or E1 link and N x DS0 (fractional T1 or E1) bundles.

configure interface bundle encapsulation

This command selects the type of protocol encapsulation for a bundle.

This entry is required for all bundles. Choose from the following:

- HDLC
- PPP
- Frame Relay

NOTE: If the bundle encapsulation is changed on an existing bundle, all packet filter, DoS, QoS, and NAT configuration for that bundle will be lost. For this reason, the user is prevented from changing bundle encapsulation. To achieve the desired changes, the user is required to delete the existing bundle and then reconfigure a new bundle with the desired encapsulation, packet filters, DoS, QoS, and NAT parameters.

Use the **configure interface bundle hdlc** command to set the HDLC parameters for a fractional T1 or E1 or single-link bundle.

Use the **configure interface bundle ppp** command to set up PPP parameters on a fractional, single-link, or multilink bundle. If you choose PPP on a multilink bundle, the Router system activates MLPPP. You can then complete the configuration of the MLPPP parameters by using the **configure interface bundle mlppp** command.

Use the **configure interface bundle fr** command to set up FR parameters for a fractional, single link, or multilink bundle.

To verify bundle encapsulation settings, use the **show interface bundle** command.

parameter	definition
encapsulation	Type of encapsulation to be used (HDLC, PPP, or frame relay).

syntax:

encapsulation < hdlc | ppp | frelay >

example:

Router/configure/interface/bundle SF_01> **encapsulation hdlc**

applicable models:

All models.

configure interface bundle fr

This command accesses next-level commands for frame relay bundle configuration.

Be sure to select frame relay encapsulation when initially creating the bundle, using the **configure interface bundle encapsulation** command.

syntax:

fr

example:

Router/configure/interface/bundle SF_01> **fr**

next-level commands

configure interface bundle fr enable
configure interface bundle fr frame_size
configure interface bundle fr intf_type
configure interface bundle fr lmi
configure interface bundle fr mfr
configure interface bundle fr pvc

applicable models:

All models.

configure interface bundle fr enable

This command accesses next-level commands for enabling or disabling an entire frame relay bundle, individual PVCs or all PVCs within a frame relay bundle.

syntax:

enable

example:

Router/configure/interface/bundle SF_01/fr> **enable**

next-level commands

configure interface bundle fr enable fragment_rfc1490
configure interface bundle fr enable interface
configure interface bundle fr enable pvc

applicable models:

All models.

configure interface bundle fr enable fragment_rfc1490

This command enables or disables rfc1490 fragmentation.

The default is enabled.

syntax:

[no] fragment_rfc1490

example:

Router/configure/interface/bundle SF_01/fr/> **enable fragment_rfc1490**

related commands:

configure interface bundle fr enable interface
configure interface bundle fr enable pvc

applicable models:

All models.

configure interface bundle fr enable interface

This command enables or disables frame relay operation on the entire bundle.

syntax:

[no] enable interface

example:

Router/configure/interface/bundle SF_01/fr> **enable interface**

related commands:

configure interface bundle fr enable fragment_rfc1490
configure interface bundle fr enable pvc

applicable models:

All models.

configure interface bundle fr enable pvc

This command enables or disables individual PVCs or all PVCs.

syntax:

[no] enable pvc pvc < n | all >

example:

Router/configure/interface/bundle SF_01/fr> **enable pvc all**

related commands:

configure interface bundle fr enable fragment_rfc1490
configure interface bundle fr enable interface

applicable models:

All models.

NOTE: The pvc number (DLCI number 16 - 1024) used in this command is for a previously established pvc.

configure interface bundle fr frame_size

This command sets the maximum frame size for a frame relay bundle.

This setting applies to all of the bundle's PVCs.

A problem can occur where certain frame sizes are dropped when DS3 FR connections come up between a Router system connected to other vendor equipment. To avoid this problem, either configure the Router router frame size to 4470 bytes or configure the other router frame size down to 4096 bytes. (Both ends should be configured for the same frame size.)

parameter	definition
bytes	Maximum frame size The range is 56 - 4096 bytes; the default is 1600.

syntax:

frame_size frame_size bytes < n >

example:

Router/configure/interface/bundle SF_01/fr> **frame_size 1024**

applicable models:

All models.

NOTE: This setting is applicable to all pvcs within a specific bundle.

NOTE: On some occasions, providing a small frame size may cause problems when a large number of pvcs are configured within the same bundle. This is because the LMI frame size is restricted to the configured frame size. Since LMI frames carry the status of all pvcs within the bundle, if the frame size is too small, the LMI packet cannot carry all of the pvcs and their status information to the remote site. Such checks are already done as part of the CLI configuration so as not to allow a very low frame size.

NOTE: Larger incoming frames are dropped. Outgoing frames are fragmented using FRC 1490/2427 to stay within the max frame size.

configure interface bundle fr intf_type

This command sets the interface type for a frame relay bundle.

parameter	definition
type	
dce	Data communications equipment (DCE)
dte	Data terminal equipment (DTE) (default)
nni	Switched network-to-network interface (NNI)

syntax:

intf_type type < dce | dte | nni >

example:

Router/configure/interface/bundle SF_01/fr> **intf_type dce**

applicable models:

All models.

configure interface bundle fr lmi

This command accesses next-level commands for configuring the local management interface (LMI) on a frame relay bundle.

This command also selects the type of LMI to be used.

parameter	definition
lmi_type	Type of LMI to be used.
ansi	ANSI T1.617 Annex D (default setting)
cisco	Cisco-compatible LMI
q933a	ITU-T Q.933 Annex A

syntax:

[no] lmi lmi_type < ansi | cisco | q933a >

example:

Router/configure/interface/bundle SF_01/fr> **lmi cisco**

applicable models:

All models.

NOTE: When LMI has been disabled on a bundle, interface type is not relevant.

configure interface bundle fr lmi dce

Sets the bundle LMI status polling interval and error threshold parameters.

In the DCE mode, the system responds to LMI polls from the remote DTE device. Use this command for both DCE and NNI interface types.

parameter	definition
n392	Error threshold (the maximum number of unreceived LMI status inquiries accepted by the system before the interface is declared down). The range is 1 - 10; the default is 3. This value must always be less than the **n393** value below.
n393	Maximum number of LMI polling intervals during which the **n392** error threshold above is counted. The range is 1 - 10; the default is 4.

syntax:

dce [n392 < n >] [n393 < n >]

example:

Router/configure/interface/bundle SF_01/fr/lmi> **dce n392 4 n393 10**

applicable models:

All models.

configure interface bundle fr lmi dte

Sets the bundle LMI status polling interval and error threshold parameters.

In the DTE mode, the system sends LMI polls to the remote DCE device. Use this command for both DTE and NNI interface types.

parameter	definition
n392	Error threshold (maximum number of unanswered LMI status inquiries accepted by the system before the interface is declared down). The range is 1 - 10; the default is 3. This value must always be less than the **n393** value below.
n393	Maximum number of LMI polling intervals during which the **n392** error threshold above is counted. The range is 1 - 10; the default is 4.
n391	Number of LMI status inquiries that pass before the system sends a full status inquiry message. The range is 1 - 255; the default is 6.

syntax:

dte [n392 < n >] [n393 < n>] [n391 < n >]

example:

Router/configure/interface/bundle SF_01/fr/lmi> **dte n392 5 n393 6 n391 20**

applicable models:

All models.

configure interface bundle fr lmi keepalive

Sets the bundle LMI polling keepalive interval for all PVCs.

The DTE generates an LMI status inquiry message once every keepalive interval and sends it to the DCE, which expects to receive status inquiries once every keepalive interval. If the DCE does not receive an inquiry during a keepalive interval, the error count advances by one.

parameter	definition
interval	The LMI polling keepalive interval The range is 5 - 255 seconds. The default is 10 seconds for DTE/NNI. The default is 15 seconds for DCE.

syntax:

keepalive interval < n >

example:

Router/configure/interface/bundle SF_01/fr/lmi> **keepalive 60**

In this example, the PVC shuts down if the bundle does not receive an LMI polling response from the network within 60 seconds.

applicable models:

All models.

configure interface bundle fr mfr

This command accesses next-level commands for configuring multilink frame relay operation on a Router system.

syntax:

configure interface bundle fr mfr

example:

Router/configure/interface/bundle/fr> **mfr**

next-level commands

configure interface bundle fr mfr ack_msg
configure interface bundle fr mfr class
configure interface bundle fr mfr diff_delay
configure interface bundle fr mfr fragment_size
configure interface bundle fr mfr hello_timer
configure interface bundle fr mfr seg_threshold

applicable models:

All models.

configure interface bundle fr mfr ack_msg

This command sets the multilink frame relay acknowledgement timer parameters.

parameter	definition
ack_timer	Time interval for which the system waits for an acknowledgement from the network device The range is 1 - 10 seconds; the default is 4 seconds.
max_retry	Maximum number of additional times the system sends an acknowledgement request to a device before dropping a link from the bundle The range is 1 - 5 seconds; the default is 2 seconds.

syntax:

mfr ack_msg ack_timer < n > max_retry < n >

example:

Router/configure/interface/bundle SF_01/fr> **mfr ack_msg ack_timer 10 max_retry 5**

applicable models:

All models.

configure interface bundle fr mfr class

This command selects the bandwidth requirement classification for a multilink frame relay bundle.

parameter	definition
class A	Bundle is up when any one T1 or E1 link is up.
class B	Bundle is up when all T1 links are up.
class C	Bundle is up when a user-specified number of T1 or E 1 links are up, per **threshold** setting.
threshold	Minimum number of active T1 or E1 links needed to activate the multilink frame relay bundle
	The range is 1 - 28.

syntax:

mfr class < A | B | C >[threshold < n >]

example:

Router/configure/interface/bundle SF_01/fr> **mfr class C threshold 4**

applicable models:

All models.

configure interface bundle fr mfr diff_delay

This command sets the differential delay for a multilink frame relay bundle.

parameter	definition
value	Tolerance, in milliseconds, to differential delay between frame relay links The range is 10 - 128; the default is 100.

syntax:

diff_delay value < n >

example:

Router>configure/interface/bundle SF_01/fr/mfr> **diff_delay 60**

applicable models:

All models.

configure interface bundle fr mfr fragment_size

This command sets the frame size above which a packet is fragmented.

Packets are intelligently fragmented when the MFR fragment size is exceeded.

parameter	definition
bytes	Maximum number of bytes in each frame The range is 56 - 4096; the default is 1500.

syntax:

mfr fragment_size frame size bytes < n >

example:

Router/configure/interface/bundle SF_01/fr> **mfr fragment_size 2048**

applicable models:

All models.

configure interface bundle fr mfr hello_timer

This command sets the hello message retransmission interval for a multilink frame relay bundle.

This is the time interval between acknowledgement requests sent to destination devices.

parameter	definition
secs	Hello time interval The range is 1 - 180 seconds; the default is 10 seconds.

syntax:

mfr hello_timer secs < n >

example:

Router/configure/interface/bundle SF_01/fr> **mfr hello_timer 30**

applicable models:

All models.

configure interface bundle fr mfr seg_threshold

This command sets the segmentation threshold for multilink frame relay frames.

The segmentation threshold can never be greater than the frame size.

parameter	definition
bytes	All packet fragments will be equal to or greater than seg_threshold. Packets less than 2 x seg_threshold will be forwarded rather than fragmented. The range is 56 - 4096; the default is 512.

syntax:

mfr seg_threshold bytes < n >

example:

Router/configure/interface/bundle SF_01/fr> **mfr seg_threshold 56**

applicable models:

All models.

NOTE: If the segmentation threshold you enter is greater than the frame size, the system disregards it.

configure interface bundle fr pvc

This command adds PVCs to a frame relay bundle and accesses commands for setting PVC parameters.

Each PVC requires a unique DLCI number for identification. To remove a PVC, type **no pvc <n>** where **<n>** is the DLCI number of that PVC.

parameter	definition
dlci	DLCI number of the PVC to be configured The range is 1 - 1022.

syntax:

[no] pvc dlci < n >

example:

Router/configure/interface/bundle SF_01/fr> **pvc 22**

next-level commands

configure interface bundle fr pvc bridge
configure interface bundle fr pvc desc
configure interface bundle fr pvc enable
configure interface bundle fr pvc icmp
configure interface bundle fr pvc ip
configure interface bundle fr pvc map
configure interface bundle fr pvc nat
configure interface bundle fr pvc policing
configure interface bundle fr pvc red
configure interface bundle fr pvc shaping
configure interface bundle fr pvc switch
configure interface bundle fr pvc vlan

applicable models:

All models.

configure interface bundle fr pvc bridge

This command supports IEEE 802 tagged frames.

parameter	definition
type	
lan	LAN packets (IEEE 802.3) to be bridged (default) Used for VLAN tagging.
vlan	VLAN packets (IEEE 802.1Q) to be bridged Used for VLAN forwarding

syntax:

[no] bridge [type < lan | vlan >]

example:

Router/configure/interface/bundle SF_01/fr/pvc 22> **bridge vlan**

applicable models:

All models.

configure interface bundle fr pvc crypto

This command configures the network type for the specified pvc on a specified Frame Relay bundle.

There is no default for this command.

parameter	definition
network_type	
trusted	Interface is part of a trusted network
untrusted	Interface is part of a untrusted network
dmz	Interface is part of a dmz network

syntax:

[no] crypto network_type < trusted | untrusted | dmz >

example:

Router1/configure/interface/bundle Dallas/fr/pvc 17> **crypto trusted**

applicable models:

OmniAccess 604

configure interface bundle fr pvc desc

This command describes the PVC.

parameter	definition
description	Name of the PVC Use up to 64 characters; enclose in quotation marks.

syntax:

desc description < " *string* " >

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **desc "San Jose Sales Office"**

related commands:

configure interface bundle
configure interface bundle fr pvc

applicable models:

All models.

configure interface bundle fr pvc enable

This command enables or disables the current PVC (DLCI number) on the selected bundle.

syntax:

[no] enable

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **enable**

applicable models:

All models.

configure interface bundle fr pvc icmp

This command accesses next-level commands for configuring ICMP messages on a frame relay PVC.

syntax:

icmp

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **icmp**

next-level commands

configure interface bundle fr pvc icmp redirect
configure interface bundle fr pvc icmp unreachable

applicable models:

All models.

configure interface bundle fr pvc icmp redirect

This command requests that a specific PVC sends ICMP redirect messages when a better route exists for the destination IP address.

syntax:

[no] icmp redirect

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **icmp redirect**

applicable models:

All models.

configure interface bundle fr pvc icmp unreachable

This command requests that a specific PVC sends ICMP unreachable messages when no route exists for a destination IP address.

syntax:

[no] icmp unreachable

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **icmp unreachable**

applicable models:

All models.

configure interface bundle fr pvc ip

This command accesses next-level commands for assigning an IP address and source forwarding address to a terminated PVC.

The command also enables and configures directed broadcasting and filtering for a terminated PVC.

syntax:

ip

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **ip**

next-level commands

configure interface bundle fr pvc ip address
configure interface bundle fr pvc ip access-group
configure interface bundle fr pvc ip directed_broadcast
configure interface bundle fr pvc ip sorce_forwarding

applicable models:

All models.

configure interface bundle fr pvc ip access-group

This command applies or removes a filter list to or from the PVC.

parameter	definition
listname	Filter rule list name to be applied.
pktdir	
in	Packets that are inbound.
out	Packets that are outbound.

syntax:

[no] ip access-group listname < *name* > pktdir < in | out >

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **ip access-group Rules_01 out**

applicable models:

All models.

configure interface bundle fr pvc ip address

This command assigns or removes a routing destination IP address and subnet mask to a PVC.

NOTE: The network 1.1.0.0 is utilized internally in the Router system. The user is prevented from configuring IP addresses within, or IP routes to, this network.

parameter	definition
IP address	IP address of the PVC
netmask	Subnet mask of the PVC
type	
broadcast	Broadcast interface

syntax:

[no] ip address < *IP address* > netmask < *subnet mask* > [type < broadcast >]

example 1:

Router/configure/interface/bundle SF_01/fr/pvc 19> ip **address 192.5.72.1 255.255.255.0 broadcast**

example 2:

Router/configure/interface/bundle SF_01/fr/pvc 19> ip **address 192.5.72.1 24 broadcast**

applicable models:

All models.

configure interface bundle fr pvc ip directed_broadcast

This command enables or disables forwarding of direct broadcasts from this interface. The default value for this command is enabled.

syntax:

[no] ip directed_broadcast

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **ip directed_broadcast**

applicable models:

All models.

configure interface bundle fr pvc ip source_forwarding

Specifies an IP address to which data on a terminated PVC will be routed.

parameter	definition
primary	Source forwarding gateway IP address
secondary	Source forwarding gateway IP address using alternate Ethernet The default is none.

syntax:

[no] ip source_forwarding primary < *IP address* > [secondary < *IP address* >]

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **ip source_forwarding 192.5.72.1**

applicable models:

All models.

configure interface bundle fr pvc map

This command assigns a static peer IP address to a PVC.

Once a static peer IP address has been assigned, inverse ARP ceases to function. If a remote IP address is changed, the static entry will not update.

Use the no option of this command to remove the static entry and automatically invoke inverse arp.

parameter	definition
ipaddr	Remote IP address of the PVC

syntax:

[no] map ipaddr < *IP address* >

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **map 10.1.100.20**

applicable models:

All models.

configure interface bundle fr pvc nat

This command accesses next-level commands for enabling network address translation (NAT) on a frame relay PVC.

Before using this command, be sure to give the PVC an IP address, using the **configure interface bundle fr pvc ip address** command.

syntax:

nat

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **nat**

next-level commands

configure interface bundle fr pvc nat address
configure interface bundle fr pvc nat enable
configure interface bundle fr pvc nat ip
configure interface bundle fr pvc nat max_entries
configure interface bundle fr pvc nat max_ports
configure interface bundle fr pvc nat pass_thru
configure interface bundle fr pvc nat port
configure interface bundle fr pvc nat reverse
configure interface bundle fr pvc nat timeout
configure interface bundle fr pvc nat trans_addr
configure interface bundle fr pvc nat trans_mode
configure interface bundle fr pvc nat unregistered

applicable models:

All models.

configure interface bundle fr pvc nat address

This command adds a local static IP address for translation to a public network (global) address.

If reverse translation is enabled on a PVC, it will also occur in the opposite direction.

parameter	definition
local_address	Local IP address
global_address	Global (public) IP address

syntax:

[no] address local_address < *IP address* > global_address < *IP address* >

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/nat> **address 10.10.1.4 150.157.99.1**

applicable models:

All models.

configure interface bundle fr pvc nat enable

This command enables or disables static and dynamic translation on a PVC.

parameter	definition
translationmode	
static	Static translation mode This is the default.
dynamic	Dynamic translation mode

syntax:

[no] enable translationmode < static | dynamic >

example:

Router/interface/bundle SF_01/fr/pvc 19/nat> **enable dynamic**

applicable models:

All models.

configure interface bundle fr pvc nat ip

This command changes the default IP address used for dynamic port translation; the default IP address is the IP address of the PVC.

parameter	definition
old_ip_address	The existing translation IP address to be changed.
new_ip_address	The IP address to which the bundle's IP address will be translated.

syntax:

ip old_ip_address < *IP address* > new_ip_address < *IP address* >

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/nat> **ip 140.141.99.28 140.110.87.14**

applicable models:

All models.

configure interface bundle fr pvc nat max_entries

This command limits the number of address translations that can occur on a PVC.

parameter	definition
max_translations	Maximum number of translations The range is 0 - 60000; the default is 60000.

syntax:

max_entries max_translations < n >

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/nat> **max_entries 120**

applicable models:

All models.

configure interface bundle fr pvc nat max_ports

This command configures the maximum ports/translations for the translation address.

parameter	definition
ipaddress	IP address
max_translations	The maximum number of translations to support.

syntax:

max_ports ipaddress < *IP address* > max_translations < n >

example:

Router/configure/interface/bundle SF_01/fr/pvc 16/nat> **max_ports 10.10.20.1 25**

applicable models:

All models.

configure interface bundle fr pvc nat pass_thru

This command enables non-translated packets to pass through on the PVC.

When you enable pass_thru, all incoming and outgoing packets without translation table entries are passed through the system unchanged. This feature is factory-enabled, however, you can disable it for additional security.

syntax:

[no] pass_thru

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/nat> **no pass_thru**

applicable models:

All models.

configure interface bundle fr pvc nat pass-thru-multicast

This command enables multicast packets to pass through on the PVC.

syntax:

[no] pass-thru-multicast

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/nat> **no pass-thru-multicast**

applicable models:

All models.

configure interface bundle fr pvc nat port

This command adds a static TCP or UDP address and port to the NAT table.

parameter	definition
protocol_type	
tcp	TCP protocol
udp	UDP protocol
local_address	Local IP address to translate
local_port	Local IP address of the local port
	The range is 1 - 65535.
global_address	Global address
global_port	Global IP address of the global port
	The range is 1 - 65535.

syntax:

[no] port protocol_type < tcp | udp > local_address < *IP address* > local_port < n > global_address < *IP address* > global_port < n >

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/nat/ port> **tcp 10.1.100.4 50 140.141.99.30 100**

applicable models:

All models.

configure interface bundle fr pvc nat reverse

This command enables reverse NAT functionality, which occurs on incoming packets from the public network.

This translates the global addresses and ports in the PVC's translation table to local addresses and ports.

syntax:

[no] reverse

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/nat> **reverse**

applicable models:

All models.

configure interface bundle fr pvc nat timeout

This command configures the timeout value for dynamic translation entries.

Router systems are factory-configured so that all dynamic TCP translation entries that have been inactive for two hours are deleted. All dynamic UDP entries that are inactive for one minute are deleted.

parameter	definition
protocol_type	
tcp	TCP protocol
udp	UDP protocol
seconds	Timeout interval, in seconds
	The range is 60 - 86400; the default for TCP is 7200 seconds (2 hours). The default for UDP is 60.

syntax:

timeout protocol_type < tcp | udp > [seconds < n >]

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/nat> **timeout tcp 3600**

applicable models:

All models.

configure interface bundle fr pvc nat trans_addr

This command adds or deletes a NAT translation port IP address for a specific frame relay PVC.

parameter	definition
ip_address	IP address of a translation port

syntax:

[no] trans_addr ip_address < *IP address* >

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/nat> **trans_addr 101.2.4.9**

applicable models:

All models.

configure interface bundle fr pvc nat trans_mode

This command sets the translation mode for a specific frame relay PVC.

parameter	definition
mode	
overflow	Overflow mode (default)
round_robin	Round-robin mode

syntax:

[no] trans_mode [mode < overflow | round_robin >]

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/nat> **trans_mode round_robin**

applicable models:

All models.

configure interface bundle fr pvc nat unregistered

This command configures the system to translate only unregistered IP addresses on a PVC.

If you enable this option, only the packets with unregistered source addresses are translated. The unregistered local addresses are those within the following ranges:

- 10.0.0.0 to 10.255.255.255
- 172.16.0.0 to 172.31.255.255
- 192.168.0.0 to 192.168.255.255

If this function is disabled, any local address (registered or unregistered) can be translated by adding the appropriate entries to the translation table or enabling dynamic port translation as previously described.

If this function is enabled, all registered local addresses will pass through without any packet modifications.

syntax:

[no] unregistered

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/nat> **unregistered**

applicable models:

All models.

configure interface bundle fr pvc policing

This command enables or disables traffic policing (incoming traffic) on a PVC.

Policing sets the committed information rate and data burst parameters that control the data flow on the PVC in the incoming direction. Policing is automatically enabled on all PVCs of all frame relay bundles.

parameter	definition
cir	Committed information rate for LMI, in bits per second
	The default is the bundle bandwidth rate.
bc	Maximum committed (guaranteed) transmission burst size for the LMI PVC, in bits
	The default is the bundle bandwidth rate.
	Generally, this value exceeds **cir** and may be a multiple of that value.
be	Excess burst size (number of bits in excess of **bc** value); non-guaranteed
	The default is 0.
de	Enable discard-eligible (DE) bit on a PVC.
	The default is off.
	This setting allows lower-priority PVCs to designate their traffic as eligible for discard during periods of heavy congestion.

syntax:

[no] policing [cir < n >] [bc < n >] [be < n >] [de < de >]

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **policing cir 128000 bc 256000 be 32000 de**

applicable models:

All models.

NOTE: Non-committed PVCs are created by setting the PVC CIR to 0.

configure interface bundle fr pvc red

This command accesses next-level commands for configuring RED on frame relay PVCs.

RED is active by default on the Router system and applies only to traffic being passed from the Ethernet port to the WAN. RED parameters can only be modified after configuring an IP address.

syntax:

[no] red

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **red**

next-level commands

configure interface bundle fr pvc red tx_min_thresh
configure interface bundle fr pvc red tx_max_thresh
configure interface bundle fr pvc red wq_bias_factor

applicable models:

All models.

configure interface bundle fr pvc red tx_max_thresh

This command sets the maximum number of buffers in a PVC queue.

If this value is exceeded during periods of traffic congestion, RED drops all packets until the queue size drops below the selected value.

parameter	definition
value	Default maximum average queue length per PVC. The range is 10 - 511; the default is 15.

syntax:

tx_max_thresh value < n >

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/red> **tx_max_thresh 14**

applicable models:

All models.

NOTE: Using a value of less than 10 for the minimum threshold is not recommended. Doing so could result in some packets being dropped.

NOTE: The maximum threshold should be set greater than the minimum threshold. For better performance, the maximum threshold should be set to at least twice the minimum threshold setting.

configure interface bundle fr pvc red tx_min_thresh

This command sets the minimum number of buffers as the threshold at which RED will begin dropping packets.

If this value is exceeded during periods of traffic congestion, RED starts dropping packets based on the wq_bias_factor setting.

parameter	definition
value	The threshold at which RED begins dropping packets, stated as the minimum number of buffers in the PVC's output queue. The range is 3 - 7; the default is 5.

syntax:

tx_min_thresh value < n >

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/red> **tx_min_thresh 4**

applicable models:

All models.

configure interface bundle fr pvc red wq_bias_factor

This command sets the probability factor for packet drops when RED is enabled.

A low value increases the probability of packet drops and reduces the PVC queue size. An extremely low value may cause RED to drop more packets than during temporary traffic congestion. A high value reduces the probability of packet drops and increases the queue size, making it better for temporary traffic bursts.

parameter	definition
value	Default weighing queue bias factor per PVC. The range is 3 - 7; the default is 5.

syntax:

wq_bias_factor value < n >

example:

Router/configure/interface/bundle SF_01/fr/pvc 19/red> **wq_bias_factor 6**

applicable models:

All models.

configure interface bundle fr pvc shaping

This command enables and disables traffic shaping (outgoing traffic) on a PVC.

Shaping sets the committed information rate and data burst parameters that control the data flow on the PVC in the outgoing direction. Shaping is automatically enabled on all PVCs of all frame relay bundles.

parameter	definition
cir	Committed information rate, in bits per second
bcmax	Maximum committed (guaranteed) transmission burst size on the PVC, in bits Generally, this value exceeds **cir** and may be a multiple of that value.
bcmin	Minimum committed burst size, in bits Generally, this is a value greater than **cir** and less than **bcmax**.
be	Excess burst size (number of bits in excess of **bcmax**); non-guaranteed

syntax:

[no] shaping [cir < n >] [bcmax < n >] [bcmin < n >] [be < n >]

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **shaping cir 128000 bcmax 256000 bcmin 192000 be 64000**

applicable models:

All models.

configure interface bundle fr pvc switch

This command enables Layer 2 switching between the current PVC and another PVC on the same bundle, or between the current PVC and a PVC on another bundle.

To ensure correct switching, specify both PVC number and bundle name. Before enabling switching, configure the other bundle and associated PVC. Both of the bundles and PVCs must exist to enable switching.

parameter	definition
dlci	DLCI number of the other PVC The range is 16 - 1022.
bundle	Name of the WAN bundle to which the other PVC belongs. This entry is not required if both PVCs are on the same bundle.

syntax:

[no] switch dlci < n > bundle < *name* >

example:

Router/configure/interface/bundle SF_01/fr/pvc 19> **switch 25 Main**

applicable models:

All models.

configure interface bundle fr pvc vlan

This command accesses next-level commands to configure VLAN tagging parameters.

syntax:

vlan

example:

Router/configure/interface/bundle SF_01/fr/pvc 16> **vlan**

next-level commands

configure interface bundle fr pvc vlan router_ip_addr
configure interface bundle fr pvc vlan vlanid

applicable models:

All models.

configure interface bundle fr pvc vlan router_ip_addr

This command configures a remote router IP address for a VLAN ARP request.

syntax:

router_ip_addr ip_address < *IP address* >

example:

Router/configure/interface/bundle SF01/fr/pvc 22/vlan> **router_ip_addr 163.54.3.1**

related commands:

configure interface bundle fr pvc vlan vlanid

applicable models:

All models.

configure interface bundle fr pvc vlan vlan_ether_type

This command configures the VLAN Ethernet type.

parameter	definition
vlan_ether_type	The Ethernet type in decimal format The range is 1 - 65535; the default is 33024 (0x8100).

syntax:

[no] vlan_ether_type vlan_ether_type < n >

example:

Router/configure/interface/bundle wan1/fr/pvc 17/vlan> **vlan_ether_type 350**

applicable models:

All models.

configure interface bundle fr pvc vlan vlanid

This command configures VLAN tagging for all incoming packets for the PVC.

parameter	definition
vlanid	The VLAN id. The range is 1 - 4095.

syntax:

[no] vlanid vlanid < n >

example:

Router/configure/interface/bundle wan08/fr/pvc 27/vlan> **vlanid 1500**

related commands:

configure interface bundle fr pvc vlan router_ip_addr

applicable models:

All models.

configure interface bundle fr pvc vlan vld_ether_type

This command configures the vld Ethernet type.

parameter	definition
vld_ether_type	The Ethernet type in decimal format The range is 1 - 65536; the default is 33024 (0x8100).

syntax:

[no] vld_ether_type vld_ether_type < n >

example:

Router/configure/interface/bundle wan1/fr/pvc 17/vlan> **vld_ether_type 2009**

applicable models:

All models.

configure interface bundle fr pvc vlan vldid

This command configures the vld tag for this pvc.

parameter	definition
vldid	The vld identification number. The range is 1 - 4095.

syntax:

[no] vldid vldid < n >

example:

Router/configure/interface/bundle wan1/fr/pvc 17/vlan> **vldid 3022**

applicable models:

All models.

configure interface bundle hdlc

This command sets the mtu and keepalive parameters for an HDLC encapsulated bundle.

parameter	definition
keepalive	The link's keep-alive interval, in seconds
	The range is 0 - 120; the default is 10 seconds.
	The Router system will send messages once every designated interval to check the bundle's status.
	To turn off the keepalive, type 0**.**
packet_type	Keepalive packet type
broadcast	Keepalive will be sent as a broadcast packet
unicast	Keepalive will be sent as a unicast packet
	This is the default.
mtu	Maximum transmission unit
	This is the maximum packet size to be sent, in bytes.
	The range is 64 - 4500; the default is 1500.

syntax:

hdlc [keepalive < n >] [packet_type < broadcast | unicast >] [mtu < n >]

example 1:

Router/configure/interface/bundle SF_01> **hdlc broadcast**

example 2:

Router/configure/interface/bundle SF_01> **hdlc unicast**

example 3:

Router/configure/interface/bundle SF_01> **hdlc keepalive 20 broadcast**

example 4:

Router/configure/interface/bundle SF_01> **hdlc unicast mtu 1500**

example 5:

Router/configure/interface/bundle SF_01> **hdlc keepalive 11 unicast mtu 1000**

applicable models:

All models.

configure interface bundle hdlc_link_activate

This command allows links which have been shutdown because of excessive HDLC errors, to be brought back up without having affecting the rest of the bundle.

syntax:

hdlc_link_activate

example:

Router/configure/interface/bundle SF_01>**hdlc_link_activate**

related commands:

configure system hdlc_link_deactivate

applicable models:

All models, except the OmniAccess 601.

configure interface bundle icmp

This command accesses next-level commands that allow the system to send various types of ICMP messages for a bundle.

syntax:

icmp

example:

Router/configure/interface/bundle SF_01> **icmp**

next-level commands

configure interface bundle icmp redirect
configure interface bundle icmp unreachable

applicable models:

All models.

configure interface bundle icmp redirect

This command sends ICMP redirect messages when a better route exists for a destination IP address.

syntax:

[no] redirect

example:

Router/configure/interface/bundle SF_01/icmp> **redirect**

related commands:

configure interface bundle icmp unreachable

applicable models:

All models.

configure interface bundle icmp unreachable

This command sends ICMP unreachable messages when no route exists for a destination IP address.

syntax:

[no] icmp unreachable

example:

Router/configure/interface/bundle SF_01> **no icmp unreachable**

related commands:

configure interface bundle icmp redirect

applicable models:

All models.

configure interface bundle ip

This command accesses next-level commands for configuring IP parameters on a bundle.

syntax:

ip

example:

Router/configure/interface/bundle SF_01> **ip**

next-level commands

configure interface bundle ip address
configure interface bundle ip access-group
configure interface bundle ip directed_broadcast
configure interface bundle ip multicast
configure interface bundle ip source_forwarding
configure interface bundle ip unnumbered

applicable models:

All models.

configure interface bundle ip access-group

This command applies a packet filtering rule set to a specific bundle.
You must first use the **configure ip access-list** commands to create the rule set.

parameter	definition
name	Name of the filtering rule set to be applied.
direction of packet transmission	
in	Packets that are inbound.
out	Packets that are outbound.

syntax:

ip access-group < *name* > < in | out >

example:

Router/configure/interface/bundle SF_01> **ip access-group Rules_01 out**

related commands:

configure interface bundle ip address
configure interface bundle ip directed_broadcast
configure interface bundle ip multicast
configure interface bundle ip source_forwarding
configure interface bundle ip unnumbered

applicable models:

All models.

configure interface bundle ip address

This command assigns an IP address and subnet mask to a bundle.

NOTE: The network 1.1.0.0 is utilized internally in the Router system. The user is prevented from configuring IP addresses within, or IP routes to, this network.

parameter	definition
ipaddress	The bundle IP address
netmask	The bundle subnet mask
type	
broadcast	Broadcast interface type

syntax:

ip address ipaddress < *IP address* > netmask < *subnet mask* > [type <broadcast >]

example:

Router/configure/interface/bundle SF_01> **ip address 10.1.100.20 255.255.1.0**

related commands:

configure interface bundle ip access-group
configure interface bundle ip directed_broadcast
configure interface bundle ip multicast
configure interface bundle ip source_forwarding
configure interface bundle ip unnumbered

applicable models:

All models.

configure interface bundle ip directed_broadcast

This command enables or disables forwarding of directed broadcasts from this interface. The default value for this command is enabled.

syntax:

[no] ip directed_broadcast

example:

For a frame relay bundle:

Router/configure/interface/bundle SF_01 pvc 19> **ip directed_broadcast**

example:

For a ppp bundle:

Router/configure/interface/bundle SF_01> **ip directed_broadcast**

related commands:

configure interface bundle ip address
configure interface bundle ip access-group
configure interface bundle ip multicast
configure interface bundle ip source_forwarding
configure interface bundle ip unnumbered

applicable models:

All models.

configure interface bundle ip multicast

This command enables IP multicasting on a bundle.

he multicast mode transparently passes data from one source to multiple remote destinations.

parameter	definition
mode	Desired multicast mode
pass	Pass all multicast data packets.
block	Block all multicast data packets.
ospfrip2	Pass OSPF and RIP (v2) packets.

syntax:

ip multicast < pass | block | ospfrip2 >

example:

Router/configure/interface/bundle SF_01> **ip multicast block**

related commands:

configure interface bundle ip address
configure interface bundle ip access-group
configure interface bundle ip directed_broadcast
configure interface bundle ip source_forwarding
configure interface bundle ip unnumbered

applicable models:

All models.

configure interface bundle ip source_forwarding

This command assigns a primary source forwarding address where all data received by a bundle will be forwarded.

Also assigns a secondary source forwarding address for configuring failover to separate devices.

parameter	definition
primary	Primary source forwarding destination address
secondary	Secondary source forwarding address

syntax:

ip source_forwarding primary < *IP address* > secondary < *IP address* >

example:

Router/configure/interface/bundle SF_01> **ip source_forwarding 10.2.200.3 10.2.200.4**

related commands:

configure interface bundle ip address
configure interface bundle ip access-group
configure interface bundle ip directed_broadcast
configure interface bundle ip multicast
configure interface bundle ip unnumbered

applicable models:

All models.

configure interface bundle ip unnumbered

This command sets the Ethernet port IP address as a source address for routing and forwarding updates and packets from the WAN.

You can also use **configure interface bundle ipmux unnumbered**.

syntax:

ip unnumbered < ethernet0 | ethernet1 >

example:

Router/configure/interface/bundle SF_01> **ip unnumbered ethernet0**

related commands:

configure interface bundle ip address
configure interface bundle ip access-group
configure interface bundle ip directed_broadcast
configure interface bundle ip multicast
configure interface bundle ip source_forwarding

applicable models:

All models.

configure interface bundle ipmux

This command accesses next-level commands for assigning an IP address, configuring source forwarding, or enabling IP unnumbered link on a bundle.

syntax:

ipmux

example:

Router/configure/interface/bundle SF_01> **ipmux**

next-level commands

configure interface bundle ipmux address
configure interface bundle ipmux source_forwarding
configure interface bundle ipmux unnumbered

applicable models:

All models.

configure interface bundle ipmux address

This command assigns a source IP address and subnet mask to a bundle.
You can also use the **configure interface bundle ip address** for this purpose.

parameter	definition
address	The source IP address
netmask	The source subnet mask address

syntax:

ipmux address address < *IP address* > netmask < *subnet mask* >

example:

Router/configure/interface/bundle SF_01> **ipmux address 129.1.1.3 255.0.0.0**

related commands:

configure interface bundle ipmux source_forwarding
configure interface bundle ipmux unnumbered

applicable models:

All models.

configure interface bundle ipmux source_forwarding

This command assigns a destination to which data received by a bundle will be forwarded. You can also use the **configure interface bundle ip source_forwarding** command for this purpose.

parameter	definition
primary	Source forwarding gateway IP address
secondary	Source forwarding gateway IP address using alternate Ethernet The default is none.

syntax:

ipmux source_forwarding primary < *IP address* > [secondary < *IP address* >]

example:

Router/configure/interface/bundle SF_01> **ipmux source_forwarding 192.5.72.1**

related commands:

configure interface bundle ipmux address
configure interface bundle ipmux unnumbered

All models.

configure interface bundle ipmux unnumbered

This command configures the Ethernet port IP address as the source address for routing and forwarding packets to and from the bundle.

syntax:

ipmux unnumbered < ethernet0 | ethernet1 >

example:

Router/configure/interface/bundle SF_01> **ipmux unnumbered ethernet0**

related commands:

configure interface bundle ipmux address
configure interface bundle ipmux source_forwarding

applicable models:

All models.

configure interface bundle link

This command accesses next-level commands for assigning WAN interfaces to a bundle.

Be sure to configure each link designated for service by using the **configure module** commands.

syntax:

link

example:

Router/configure/interface/bundle SF_01> **link**

next-level commands

configure interface bundle link ct3
configure interface bundle link e1
configure interface bundle link t1
configure interface bundle link t3
configure interface bundle link ussi

applicable models:

All models.

configure interface bundle link e1

This command assigns one or more E1 links to a WAN bundle.

parameter	definition
e1	E1 link to be assigned The range is 1 - 16, depending on the Router system.
DS0	DS0 channels to be assigned to a fractional E1 bundle The range is 1 - 31.
speed	Transmission speed of all DS0 channels in the bundle (56 or 64 kbps). If this parameter is not entered, all DS0 channels will operate at 64 kbps.
inverted_data	Whether or not to invert the data on all DS0 channels (optional entry). If **inverted_data** is not entered, the data will not be inverted on any DS0 channels.

syntax:

[no] link e1 < n > [: < n >] [speed < 56 | 64 >] [inverted_data]

example 1:
RouterE/configure/interface/bundle Tokyo_6> **link e1 1**

This example assigns E1 WAN link 1 to a bundle.

example 2:
RouterE/configure/interface/bundle Tokyo_6> **link e1 1-4 inverted data**

This example assigns E1 links 1 through 4 to a multilink bundle, with data inversion to occur on all channels. The transmission speed on all channels will be 64 kbps.

example 3:
RouterE/configure/interface/bundle Tokyo_6> **link e1 1:1-6,10-15 speed 56**

This example assigns DSO channels 1 to 6 and 10 to 15 of E1 link 1 to a fractional E1 bundle. All DSO channels on this bundle will operate at 56 kbps without data inversion.

related commands:

configure interface bundle link ussi

applicable models:

OmniAccess 601, OmniAccess 602E

configure interface bundle link t1

This command assigns one or more T1 links to a WAN bundle.

parameter	definition
t1	T1 link to be assigned The range is 1 - 16, depending on the Router system.
DS0	DS0 channels to be assigned to a fractional T1 bundle (1 - 24).
speed	Transmission speed of all DS0 channels in the bundle (56 or 64 kbps) If this parameter is not entered, all DS0 channels will operate at 64 kbps.
inverted_data	Whether or not to invert the data on all DS0 channels (optional entry) If **inverted_data** is not entered, the data will not be inverted on any DS0 channels.

syntax:

[no] link t1 < 1 - 16 > [: < 1 - 24 >] [speed < 56 | 64 >] [inverted_data]

example 1:
Router/configure/interface/bundle SF_01> **link t1 1**

This example assigns T1 WAN link 1 to a bundle.

example 2:
Router/configure/interface/bundle SF_01> **link t1 1-4 inverted data**

This example assigns T1 links 1 through 4 to a multilink bundle, with data inversion to occur on all channels. The transmission speed on all channels will be 64 kbps.

example 3:
Router/configure/interface/bundle SF_01> **link t1 1:1-6 10-15 speed 56**

This example assigns DSO channels 1 to 6 and 10 to 15 of T1 link 1 to a fractional T1 bundle. All DSO channels on this bundle will operate at 56 kbps without data inversion.

related commands:

configure interface bundle link ct3
configure interface bundle link t3
configure interface bundle link ussi

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure interface bundle link_restore

This command restores a link that was previously dropped from a multilink T1 or E1 bundle.

This command is not needed if the bundle is configured for automatic link restoral using the **configure interface bundle restore** command.

syntax:

link_restore

example:

Router/configure/interface/bundle SF_01> **link_restore**

applicable models:

All models.

NOTE: If **link_restore** is set to manual and a T1 or E1 link is dropped from a bundle, reconfiguring link restore to automatic does not bring link back into bundle. User should restore any dropped links manually before reconfiguring for automatic link restore.

SNMP/MHU

MHU functionality allows an SNMP set to automate the client access authorizations. The SNMP set requires a client address and a "time to live" value (in seconds) to generate an access rule. When the time to live expires, the access rule denies access and the user must purchase additional connectivity time.

NOTE: When MHU is enabled, NAT is not supported in the current version of the software.

SNMP Set

The SNMP set command configures the access rule for a specific IP address for a specific time. The result is equivalent to the set_timeout command.

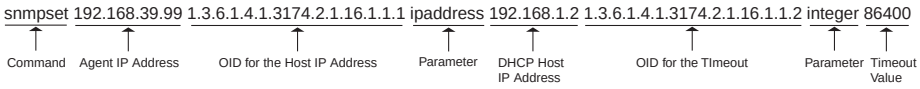
syntax:

snmpset < *agent ip address* > 1.3.6.1.4.1.3174.2.14.1.1.1 ip address < *ip address* > 1.3.6.1.4.1.3174.2.1.14.1.1.2 integer < *timeout value* >

example:

To set the access rule for the IP address 192.168.1.2 on agent 192.168.39.99 for 24 hours (86400 seconds), use the following SNMP set command:

Figure 6 SNMP Set Command Example



SNMP Get

The SNMP get command gets the value of the specific field (when indexed by the correct IP address).

syntax:

snmpget <*agent ip address* > 1.3.6.1.4.1.3174.2.14.1.1.X < *ip address* >

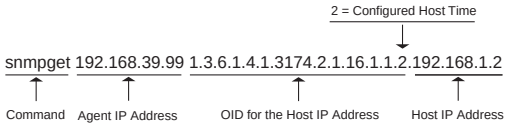
The possible values of X are:

- DHCP host IP address
- Configured host time
- Static host IP address
- Remaining host time

example:

To get the configured host time for host 192.168.1.2 on agent 192.168.39.99, use the following SNMP get command:

Figure 7 SNMP Get Command Example



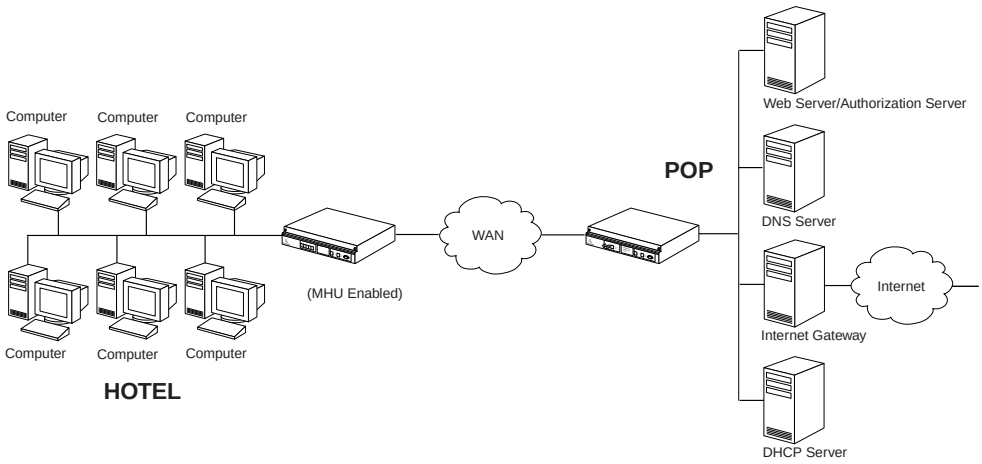
Configuring MHU

For MHU to function correctly, it requires network access to a:

- DNS server
- DHCP server
- Web server or authorization server

The DNS server replies to DNS queries that are redirected or forwarded by the MHU unit. The DHCP server performs DHCP address negotiations for both DHCP clients and DHCP proxy clients. The web server/authorization server provides IP addresses for redirected HTTP packets and sends the forced-first-page. The authorization server uses SNMP to authorize a client for network access.

Figure 8 A Typical Deployment



The example in Figure 3 represents the case when a different server is used for each required function. The most cost-effective method would be to have all server functions (DHCP, DNS, web, and authorization) in one server, thus reducing both equipment cost and administration effort.

configure interface bundle mhu

This command configures MHU functionality on a per WAN interface basis.

A WAN interface that is enabled for MHU will be the interface from which all upstream traffic will sent. A WAN interface configured for MHU must also have MHU enabled to implement this feature.

syntax:

[no] mhu

example:

Router/configure/interface/bundle SanDiego> **mhu**

next-level commands

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu access_srvr_addr

This command configures the IP address of the designated upstream access server for MHU functionality.

The access server is a single server that hosts DHCP services, DNS services, and access or authorization services required with the MHU function. If the access server is configured, there is no need to configure individual DHCP, DNS, or authorization servers.

syntax:

[no] access_srvr_addr < *IP address* >

example:

Router/configure/interface/bundle SanDiego/mhu> **access_srvr_addr 10.1.1.99**

related commands:

configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu auth_srvr_addr

This command configures the IP address of stand-alone upstream authorization server.

The authorization server is the system that will grant access to the downstream workstations through SNMP set commands.

syntax:

[no] auth_srvr_addr < *IP address* >

example:

Router/configure/interface/bundle SanDiego/mhu> **auth_srvr_addr 10.1.1.99**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu auth_srvr_port

This command configures the authorization server port.

parameter	definition
port	Port for the authorization server The range is 1 - 65535; the default is defined in the packet for HTTP traffic.

syntax:

auth_srvr_port port < n >

example:

Router/configure/interface/bundle SanDiego/mhu> **auth_srvr_port 110**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu cleanup_timer

This command configures the amount of time to expire before completely cleaning up a MHU host entry that no longer has access rights to upstream services.

parameter	definition
timer	Time in minutes The the range is 120 - 3600; the default is 10 minutes (600 seconds).

syntax:

[no] cleanup_timer timer < n >

example:

Router/configure/interlace/bundle SanDiego/mhu> **cleanup_timer 60**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu community

This command enables the default SNMP community public.

syntax:

community

example:

Router/configure/interface/bundle SanDiego/mhu> **community**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu delete_all

This command deletes all MHU host entries in the table.

syntax:

delete_all

example:

Router/configure/interface/bundle SanDiego/mhu> **delete_all**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu delete_entry

This command deletes specific MHU host entries.

parameter	definition
dhcp address	IP address of the MHU host to be deleted from the table.

syntax:

delete_entry dhcp address < *IP address* >

example:

Router/configure/interface/bundle SanDiego/mhu> **delete_entry 10.1.1.9**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu dhcp_srvr_addr

This command configures the IP address of the stand-alone upstream DHCP server.

This would be the DHCP server providing IP addresses to the downstream DHCP configured workstations.

syntax:

[no] dhcp_srvr_addr < *IP address* >

example:

Router/configure/interface/bundle SanDiego/mhu> **dhcp_srvr_addr 10.1.1.99**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu dns_srvr_addr

This command configures the IP address of the upstream stand-alone DNS server. This would be the DNS server providing name resolution to the downstream workstations.

syntax:

[no] dhcp_srvr_addr < *IP address* >

example:

Router/configure/interface/bundle SanDiego/mhu> **dns_srvr_addr 10.1.1.99**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu dns_srvr_port

This command configures the DNS server port.

parameter	definition
port	Port for the DNS server The range is 1 - 65535; the default is typically defined in the packet for DNS traffic.

syntax:

dns_srve_port port < n >

example:

Router Router/configure/interface/bundle SanDiego/mhu> **dns_srvr_port 80**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu enable

This command enables or disables MHU on a WAN configured for MHU functionality.

Before enabling MHU, it must first be configured on the appropriate authentication server, DHCP server, DNS server, or Ethernet interface. Disabling the MHU function will cause all MHU host entries to be deleted. By default the MHU function is disabled.

syntax:

[no] enable

example:

Router/configure/interface/bundle SanDiego/mhu> **enable**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu ethernet

This command configures an Ethernet interface for MHU mode.

parameter	definition
interface number	Identifies the Ethernet interface to be configured for MHU funtionality Enter 0 or 1.

syntax:

[no] ethernet interface number < 0 | 1>

example:

Router/configure/interlace/bundle SanDiego/mhu> **ethernet 1**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

configure interface bundle mhu redirect

This command enables or disables the redirect capability.

This feature redirects all HTTP, DNS, and DHCP traffic incoming for the local workstations to the configured upstream systems. By default, the redirection feature is enabled when MHU is enabled. If the redirection feature is not enabled, packet redirection must occur at the POP site.

syntax:

[no] redirect

example:

Router/configure/interlace/bundle/SanDiego/mhu> **redirect**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu send_client_info
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu send_client_info

This command appends the client's actual IP address and Ethernet MAC address at the end of an HTTP request packet during redirection.

The default is off.

syntax:

send_client_info

example:

Router/configure/interface/bundle SanDiego/mhu> **send_client_info**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_router_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu send_router_info

This command enables or disables appending the specified router's IP address during redirection.

When this option is enabled (with each redirected HTTP request only), the system will append the following string (that contains the router's configured IP address) to the end of the HTTP request packet.

"...router-ip: 192.168.23.200."

The IP address appended to the end of the HTTP request packet must be an IP address assigned to one of the router's local interfaces and be reachable by the authorization server. This option is useful for identifying the MHU router when issuing the SNMP set commands.

By default this command is disabled.

parameter	definition
router_address	IP address of router

syntax:

[no] send_router_info router_address < *IP address* >

example 1:
Router/configure/interface/bundle SanDiego/mhu> **send_router_info 192.168.23.200**

This example enables router information in the HTTP redirect packet.

example 2:
Router/configure/interface/bundle SanDiego/mhu> **no send_router_info 192.168.23.200**

This example disables router information in the HTTP redirect packet.

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_client_info
configure interface bundle mhu set_timeout

applicable models:

All models.

configure interface bundle mhu set_timeout

This command configures the timeout (TTL) for a specific host entry.

The timeout will ensure that the host will not be given Internet access beyond the specified time.

parameter	definition
dhcp address	The IP address of the host to be configured with a timeout parameter.
timeout	The timeout in seconds
-1	Infinite timeout value
0	Deletes timeout value
n	Enter a specific timeout value (number)

syntax:

set_timeout dhcp address < *IP address* > timeout < n >

example:

Router/configure/interface/bundle SanDiego/mhu> **set_timeout 1200**

related commands:

configure interface bundle mhu access_srvr_addr
configure interface bundle mhu auth_srvr_addr
configure interface bundle mhu auth_srvr_port
configure interface bundle mhu cleanup_timer
configure interface bundle mhu community
configure interface bundle mhu delete_all
configure interface bundle mhu delete_entry
configure interface bundle mhu dhcp_srvr_addr
configure interface bundle mhu dns_srvr_addr
configure interface bundle mhu dns_srvr_port
configure interface bundle mhu enable
configure interface bundle mhu ethernet
configure interface bundle mhu redirect
configure interface bundle mhu send_router_info
configure interface bundle mhu send_client_info

applicable models:

All models.

configure interface bundle mlppp

This command sets the MLPPP parameters for a PPP encapsulated bundle.

You must first choose PPP as the encapsulation type, using the **configure interface bundle encapsulation** command. When you choose PPP in a multilink bundle, the Router system activates MLPPP automatically.

parameter	definition
mrru	Maximum receive reconstructed unit (minimum, default, and maximum number of octets in the information fields of reassembled packets)
	Default range is 1500 - 1524 - 8192. The range for any particular field is 1500 - 8192.
sequence	MLPPP sequence number length (short is 12 bits, long is 24 bits)
	The default is long.
seg_threshold	All packet fragments will be equal to or greater than seg_threshold. Packets less than 2 x seg_threshold will be forwarded on a single T1 or E1 rather than fragmented across multiple T1s or E1s.
	The range is 64 - 4500; the default is 512.
differential_delay	Tolerance, in milliseconds, to differential delay between links
	The range is 0 - 128; the default is 128 milliseconds.
discriminator	IP address of the MLPPP discriminator
	The default is the bundle's IP address.

syntax:

mlppp [mrru < min - default - max >] [sequence < short | long >] [seg_threshold < n >] [differential_delay < n >] [discriminator < *IP address* >]

example:

Router/configure/interface/bundle SF_01> **mlppp mrru 1200-1500-1800 sequence short seg_threshold 1400 differential_delay 20 discriminator 10.1.100.5**

applicable models:

All models.

configure interface bundle nat

This command accesses next-level commands for enabling network address translation (NAT) on a WAN bundle.

Before using this command, be sure to give the bundle an IP address, using the **configure interface bundle ip address** command.

syntax:

[no] nat

example:

Router/configure/interface/bundle SF_01> **no nat**

next-level commands

configure interface bundle nat address
configure interface bundle nat enable
configure interface bundle nat ip
configure interface bundle nat max_entries
configure interface bundle nat pass_thru
configure interface bundle nat max_ports
configure interface bundle nat port
configure interface bundle nat reverse
configure interface bundle nat timeout
configure interface bundle nat trans_addr
configure interface bundle nat trans_mode
configure interface bundle nat unregistered

applicable models:

All models.

configure interface bundle nat address

This command adds a local static IP address for translation to a public network (global) address.

If reverse translation is enabled on that bundle, it will also occur in the opposite direction. To remove a static address from the translation table, type **no address** followed by the local and global IP addresses to be removed.

parameter	definition
local_address	Local IP address
global_address	Global (public) address

syntax:

[no] address local_address < *IP address* > global_address < *IP address*>

example:

Router/configure/interface/bundle SF_01/nat> **address 10.10.1.5 150.157.99.2**

The above command entry will translate local IP address 10.10.1.5 on a private network to global address 150.157.99.2 on the public network (Internet). If reverse translation is also enabled, the global address will also be translated back to the local address when data is received from the public network.

related commands:

configure interface bundle nat enable
configure interface bundle nat ip
configure interface bundle nat max_entries
configure interface bundle nat pass_thru
configure interface bundle nat max_ports
configure interface bundle nat port
configure interface bundle nat reverse
configure interface bundle nat timeout
configure interface bundle nat trans_addr
configure interface bundle nat trans_mode
configure interface bundle nat unregistered

applicable models:

All models.

configure interface bundle nat enable

This command enables static address and port translation, and enables dynamic port translation on a bundle.

Static translation is factory-enabled on all Router systems, and dynamic translation is factory-disabled. Type **no enable static** or **no enable dynamic** to disable a previously enabled form of translation.

parameter	definition
static	Static address and port translation
dynamic	Dynamic port translation

syntax:

[no] enable < static | dynamic >

example:

Router/configure/interface/bundle SF_01/nat> **enable dynamic**

related commands:

configure interface bundle nat address
configure interface bundle nat ip
configure interface bundle nat max_entries
configure interface bundle nat pass_thru
configure interface bundle nat max_ports
configure interface bundle nat port
configure interface bundle nat reverse
configure interface bundle nat timeout
configure interface bundle nat trans_addr
configure interface bundle nat trans_mode
configure interface bundle nat unregistered

applicable models:

All models.

configure interface bundle nat ip

This command is used to change the dynamic port translation address.

parameter	definition
old_ip_address	The old translation IP address
new_ip_address	The new translation IP address

syntax:

ip old_ip_address <*IP address* > new_ip_address < *IP address*>

example:

Router/configure/interface/bundle SF_01/nat> **ip 140.141.99.29 140.110.100.10**

related commands:

configure interface bundle nat address
configure interface bundle nat enable
configure interface bundle nat max_entries
configure interface bundle nat pass_thru
configure interface bundle nat max_ports
configure interface bundle nat port
configure interface bundle nat reverse
configure interface bundle nat timeout
configure interface bundle nat trans_addr
configure interface bundle nat trans_mode
configure interface bundle nat unregistered

applicable models:

All models.

configure interface bundle nat max_entries

This command limits the number of address translations that can occur on a bundle.

parameter	definition
max_translations	Maximum number of translations

syntax:

max_entries max_translations < n >

example:

Router/configure/interface/bundle SF_01/nat> **max_entries 100**

related commands:

configure interface bundle nat address
configure interface bundle nat enable
configure interface bundle nat ip
configure interface bundle nat pass_thru
configure interface bundle nat max_ports
configure interface bundle nat port
configure interface bundle nat reverse
configure interface bundle nat timeout
configure interface bundle nat trans_addr
configure interface bundle nat trans_mode
configure interface bundle nat unregistered

applicable models:

All models.

configure interface bundle nat pass_thru

This command enables pass-thru on a bundle.

When you enable pass_thru, all incoming and outgoing packets without translation table entries are passed through the system unchanged. This function is factory-enabled; however, you can disable it for additional security. To do this, type **no pass_thru**.

syntax:

[no] pass_thru

example:

Router/configure/interface/bundle SF_01/nat> **no pass_thru**

related commands:

configure interface bundle nat address
configure interface bundle nat enable
configure interface bundle nat ip
configure interface bundle nat max_entries
configure interface bundle nat pass-thru-multicast
configure interface bundle nat max_ports
configure interface bundle nat port
configure interface bundle nat reverse
configure interface bundle nat timeout
configure interface bundle nat trans_addr
configure interface bundle nat trans_mode
configure interface bundle nat unregistered

applicable models:

All models.

configure interface bundle nat pass-thru-multicast

This command enables multicast pass-thru on a bundle.

syntax:

[no] pass-thru-multicast

example:

Router/configure/interface/bundle SF_01/nat> **no pass-thru-multicast**

related commands:

configure interface bundle nat address
configure interface bundle nat enable
configure interface bundle nat ip
configure interface bundle nat max_entries
configure interface bundle nat pass-thru
configure interface bundle nat max_ports
configure interface bundle nat port
configure interface bundle nat reverse
configure interface bundle nat timeout
configure interface bundle nat trans_addr
configure interface bundle nat trans_mode
configure interface bundle nat unregistered

applicable models:

All models.

configure interface bundle nat max_ports

This command configures the maximum number of ports/translations to be supported for the translation address.

parameter	definition
ip address	IP address Enter an IP address.
max_translations	The maximum number of translations to support Enter a number.

syntax:

max_ports [no] ip address < *IP address* > max_translations < n >

example:

Router/configure/interface/bundle SF_01/nat> **max_ports 100.10.32.1 42**

related commands:

configure interface bundle nat address
configure interface bundle nat enable
configure interface bundle nat ip
configure interface bundle nat max_entries
configure interface bundle nat pass_thru
configure interface bundle nat port
configure interface bundle nat reverse
configure interface bundle nat timeout
configure interface bundle nat trans_addr
configure interface bundle nat trans_mode
configure interface bundle nat unregistered

applicable models:

All models.

configure interface bundle nat port

This command adds static ports to a bundle for translation.
Static ports may be added for both TCP- and UDP-protocol ports.

parameter	definition
protocol_type	
tcp	TCP protocol
udp	UDP protocol
local_address	Local address
local_port	Local port of the local address
	The range is 1 - 65535.
global_address	Global address
global_port	Global port of the global address
	The range is 1 - 65535.

syntax:

[no] protocol_type < tcp | udp > local_address < *local IP address* > local_port < n > global_address < *global IP address* > global_port < n >

example:

Router/configure/interface/bundle SF_01/nat/port> **tcp 100.10.32.1 42 100.141.99.30 63**

related commands:

configure interface bundle nat address
configure interface bundle nat enable
configure interface bundle nat ip
configure interface bundle nat max_entries
configure interface bundle nat pass_thru
configure interface bundle nat max_ports
configure interface bundle nat reverse
configure interface bundle nat timeout
configure interface bundle nat trans_addr
configure interface bundle nat trans_mode
configure interface bundle nat unregistered

applicable models:

All models.

configure interface bundle nat reverse

This command enables or disables reverse NAT, which occurs on incoming packets from the public network.

This translates the global addresses and ports in the bundle's translation table to local addresses and ports. This function is factory-disabled on all Router systems.

syntax:

[no] reverse

example:

Router/configure/interface/bundle SF_01/nat> **reverse**

related commands:

configure interface bundle nat address
configure interface bundle nat enable
configure interface bundle nat ip
configure interface bundle nat max_entries
configure interface bundle nat pass_thru
configure interface bundle nat max_ports
configure interface bundle nat port
configure interface bundle nat timeout
configure interface bundle nat trans_addr
configure interface bundle nat trans_mode
configure interface bundle nat unregistered

applicable models:

All models.

configure interface bundle nat timeout

This command sets the NAT timeout interval for dynamic translation entries for a bundle.

Router systems are factory-configured so that all dynamic TCP translation entries that have been inactive for two hours are deleted. Also, all dynamic UDP entries inactive for one minute are deleted.

parameter	definition
protocol_type	
tcp	TCP protocol
udp	UDP protocol
seconds	Number of seconds for timeout The default for TCP is 7200. The default for UDP is 60. The valid range is 60 - 86400.

syntax:

timeout protocol_type < tcp | udp > [seconds < n >]

example:

Router/configure/interface/bundle SF_01/nat> **timeout udp 3600**

related commands:

configure interface bundle nat address
configure interface bundle nat enable
configure interface bundle nat ip
configure interface bundle nat max_entries
configure interface bundle nat pass_thru
configure interface bundle nat max_ports
configure interface bundle nat port
configure interface bundle nat reverse
configure interface bundle nat trans_addr
configure interface bundle nat trans_mode
configure interface bundle nat unregistered

applicable models:

All models.

configure interface bundle nat trans_addr

This command adds or deletes the NAT translation port IP address for a specific bundle.

parameter	definition
IP address	Dynamic port translation Enter an IP address. The range is 1 - 10; the default is 1.

syntax:

[no] trans_addr < *IP address* >

example:

Router/configure/interface/bundle SF_01/nat> **trans_addr 101.2.4.9**

related commands:

configure interface bundle nat address
configure interface bundle nat enable
configure interface bundle nat ip
configure interface bundle nat max_entries
configure interface bundle nat pass_thru
configure interface bundle nat max_ports
configure interface bundle nat port
configure interface bundle nat reverse
configure interface bundle nat timeout
configure interface bundle nat trans_mode
configure interface bundle nat unregistered

applicable models:

All models.

configure interface bundle nat trans_mode

This command sets the NAT translation mode to either overflow or round-robin for a specific bundle.

parameter	definition
mode	
overflow	Overflow mode (default)
round_robin	Round-robin mode

syntax:

[no] trans_mode [mode < overflow | round_robin >]

example:

Router/configure/interface/bundle SF_01/nat> **trans_mode round_robin**

related commands:

configure interface bundle nat address
configure interface bundle nat enable
configure interface bundle nat ip
configure interface bundle nat max_entries
configure interface bundle nat pass_thru
configure interface bundle nat max_ports
configure interface bundle nat port
configure interface bundle nat reverse
configure interface bundle nat timeout
configure interface bundle nat trans_addr
configure interface bundle nat unregistered

applicable models:

All models.

configure interface bundle nat unregistered

This command configures the system to translate only unregistered IP addresses on a bundle.

If you enable this option, only the packets with unregistered source addresses are translated. The unregistered local addresses are those within the following ranges:

- 10.0.0.0 to 10.255.255.255
- 172.16.0.0 to 172.31.255.255
- 192.168.0.0 to 192.168.255.255

To disable this function, type **no unregistered**. If you do this, any local address (registered or unregistered) can be translated by adding the appropriate entries to the translation table or enabling dynamic port translation as previously described.

If this function is enabled, all registered local addresses will pass through without any packet modifications.

syntax:

[no] unregistered

example:

Router/configure/interface/bundle SF_01/nat> **unregistered**

related commands:

configure interface bundle nat address
configure interface bundle nat enable
configure interface bundle nat ip
configure interface bundle nat max_entries
configure interface bundle nat pass_thru
configure interface bundle nat max_ports
configure interface bundle nat port
configure interface bundle nat reverse
configure interface bundle nat timeout
configure interface bundle nat trans_addr
configure interface bundle nat trans_mode

applicable models:

All models.

configure interface bundle peer_addr

This command assigns the IP address of the Router system PPP peer.

Any IP address previously provided by a remote device will be ignored.

syntax:

[no] peer_addr ipaddress < *IP address* >

example:

Router/configure/interface/bundle Toronto> **peer_addr 105.100.10.1**

applicable models:

All models.

configure interface bundle ppp

This command configures the PPP parameters for a single- or fractional-link PPP bundle.

parameter	definition
mtu	Minimum, default, and maximum packet sizes to be sent, separated by dashes. For example, **100 - 250 - 1000** selects packets ranging from 100 to 1000 bytes, with a default size of 250 bytes. The default is 64 - 1500 - 4500.
mru	Minimum, default, and maximum packet sizes to be received, separated by dashes. For example, **100 - 200 - 500** selects packets ranging from 100 to 500 bytes, with a default size of 200 bytes. The default is 64 - 1500 - 4500.
magic_check	
enable	Enables magic number (default).
disable	Disables magic number.

syntax:

ppp [mtu < min - default - max >] [mru < min - default - max >] [magic_check < enable | disable >]

example:

Router/configure/interface/bundle SF_01> **ppp mtu 100-250-1000 mru 100-200-500 magic_check enable**

applicable models:

All models.

configure interface bundle qos

This command accesses next-level commands for configuring Quality of Service (QoS) on bundles.

syntax:

qos

example:

Router/configure/interface/bundle SF_01> **qos**

next-level commands

configure interface bundle qos add_class
configure interface bundle qos class
configure interface bundle qos delete_all
configure interface bundle qos delete_class
configure interface bundle qos enable

applicable models:

All models.

configure interface bundle qos add_class

This command creates a new QoS class.

The **class_name**, **parent**, **cr**, and **br** parameters must be entered; the **other** parameters are optional. Optional parameters not entered during the creation of a QoS class may be entered using the **configure interface bundle qos class** command.

parameter	definition
class_name	The name assigned to a new QoS class. The maximum number of characters allowed is 19.
parent	Name of the parent class; QoS classes are defined hierarchically. To add to the root, specify either **root-in** or **root-out.**
cr	Committed Rate (entered in Kbps). Minimum value = 1 Kbps. Maximum value = interface bandwidth. The sum of CRs of all classes at a specific level cannot exceed BR. Supported for outbound traffic only.
cr_percent	Committed Rate as a percentage of interface bandwidth. Valid Range(s) : 1 - 100
br	Burst Rate (entered in Kbps). Minimum value = 1 Kbps. Maximum value = interface bandwidth or the BR value of the parent class, whichever is lesser. BR >= CR for a class. Supported for outbound traffic only.
br_percent	Burst (Peak) Rate as % of interface bandwidth Valid Range(s) : 1 - 100
priority	Scheduling priority. Valid only for leaf classes. The range is 1 - 8. Supported for outbound traffic only.
src_ip_address	Source IP address, range of source IP addresses, source subnet, or (default) to which QoS parameters will apply.
dst_ip_address	Destination IP address, range of destination IP addresses, destination subnet, or (default) to which QoS parameters will apply.
netmask	Subnet mask used to specify a source or destination subnet to which QoS parameters will be applied.
port	Port number or (default) indicating which applications QoS parameters will be applied to.
vlan_id	The VLAN ID number or range of VLAN ids to which the QoS parameters will apply. The range is 1 - 4095 or default.
dscp	DiffServ codepoint or a range of DiffServ codepoints to which the QoS parameters will apply. The range is 0 - 63.
dot1p	802.2p priority. Valid range is 0-7.
nat_ip	Specifies a NAT translation address for this class
mark_dscp	Marks all packets with a specific DiffServ codepoint The range is 0 - 63.
mark_vlan	Tags packets with specific VLAN id if tagging is enabled
mark_dot1p	Marks packets with 802.1p priority. Valid range is: 0-7.

Each class can be configured using either explicit burst and committed rates or templates. The syntax for these commands is :

syntax 1:

add_class class_name parent [cr] [cr_percent] [br] [br_percent] [priority] [src_ip_address]
[dst_ip_address] [netmask] [port] [vlan_id] [dscp] [dot1p] [nat_ip] [mark_dscp] [mark_vlan] [mark_dot1p] <cr>

syntax 2:

add_class < class_name > < parent > cr < n > br < n > [priority < n >] [port < n >]
[vlan_id < vlan id >] [dscp < n >] [nat_ip < nat_ip address >] [mark_dscp < n >] [mark_vlan < n >]

example:

Router/configure/interface/bundle SF_01/qos> **add_class eng3 root-out cr 3000 br 6000**

applicable models:

All models.

NOTE: All QoS classes created under a parent QoS class must be classified using the same classification type: source IP address, destination IP address, port number, VLAN ID, or TOS. For example: If two classes, **eng1** and **eng2**, are created under the **root-in** parent class, it is not possible to assign **eng1** QoS parameters to a source IP address and **eng2** QoS parameters to a destination IP address. They must both apply to either source IP addresses or destination IPs.

Additional Information

- Traffic conditioning is possible only for outbound classes. Therefore, cr, br, and priority parameters are valid only for outbound classes.
- The class name should be unique for the bundle.
- The parent should be an existing class or "root-in" or "root-out."
- A maximum of 256 IP addresses can be specified in a range.
- Setting any of the classification types to a special value of "default" specifies a default class unless a default class is specified, packets not matching any of the configured classes will be dropped.
- The netmask parameter is used to specify a source subnet or a destination subnet. The default value for the netmask is 255.255.255.255. For IP address ranges only the default value of the netmask applies. In other words, a range of IP subnets cannot be specified.

related commands:

configure interface bundle qos class
configure interface bundle qos delete_all
configure interface bundle qos delete_class

applicable models:

All models.

configure interface bundle qos class

This command accesses next-level commands to modify parameters of this traffic class.

parameter	definition
class_name	Name of class being modified.

syntax:

class class_name

example:

Router/configure/interface/bundle SF_01/qos> **class eng3**

next-level commands

configure interface bundle qos class add_dscp
configure interface bundle qos class add_dst_ip
configure interface bundle qos class add_port
configure interface bundle qos class add_src_ip
configure interface bundle qos class add_vlan_id
configure interface bundle qos class burst_rate
configure interface bundle qos class commited_rate
configure interface bundle qos class delete_ip_address
configure interface bundle qos class delete_dscp
configure interface bundle qos class delete_port
configure interface bundle qos class delete_vlan_id
configure interface bundle qos class enable
configure interface bundle qos class ewf
configure interface bundle qos class mark_dscp
configure interface bundle qos class nat_ip
configure interface bundle qos class mark_vlan
configure interface bundle qos class priority
configure interface bundle qos class queue_buffers
configure interface bundle qos class red
configure interface bundle qos class add_dot1p
configure interface bundle qos class br_p
configure interface bundle qos class cr_p
configure interface bundle qos class delete_dot1p
configure interface bundle qos class mark_dot1p

applicable models:

All models.

configure interface bundle qos class add_dot1p

This command adds an 802.1p priority value to the class.

parameter	definition
add_dot1p <priority value>	Priority value range is 0 - 7.

syntax:

add_dot1p dot1p_value < n >

example:

Router/configure/interface/bundle wan1/qos/class v100> add_dot1p 3

related commands:

configure interface bundle qos add_class

applicable models:

All models.

configure interface bundle qos class add_dscp

This command adds DiffServ code points to this class.

A range can also be specified. Assured forwarding code points can be specified by using keywords "af11-af43." The expedited forwarding code point can be specified as "ef." Class selector code points can be specified using "cs0-cs7."

parameter	definition
ds_codepoint	Specifies the DiffServ code point. The range is 0 - 7

syntax:

add_dscp ds_codepoint < n >

example 1:

Router/configure/interface/bundle wan1/qos/class v100> **add_dscp 11**

example 2:

Router/configure/interface/bundle wan1/qos/class c1> **add_dscp af43**

example 3:

Router/configure/interface/bundle wan1/qos/class c2> **add_dscp ef**

applicable models:

All models.

configure interface bundle qos class add_dst_ip

This command assigns a destination IP address or subnet to the specified class.

parameter	definition
ip_address	Destination IP address, range, or default to be added to the specified class.
netmask	The subnet mask The default is 32 (255.255.255.255).

syntax 1:

add_dst_ip ip_address < *IP address* > [netmask < *subnet mask* >]

syntax 2:

add_dst_ip < default >

example 1:

Router/configure/interface/bundle SF_01/qos/class eng3> **add_dst_ip 10.1.4.0 255.255.255.0**

example 2:

Router/configure/interface/bundle SF_01/qos/class eng3> **add_dst_ip default**

This example makes this class a default class.

example 3:

Router/configure/interface/bundle SF_01/qos/class eng3> **add_dst_ip 10.1.4.1- 10.1.4.20**

applicable models:

All models.

configure interface bundle qos class add_port

This command assigns an application port to the specified class.

To assign multiple ports to a class, this command must be executed multiple times.

parameter	definition
port	Number of the application port to be added to a QoS class. The range is 1 - 65535 or default.

syntax:

add_port port < n | default >

example 1:

Router/configure/interface/bundle SF_01/qos/class eng3> **add_port 80**

example 2:

Router/configure/interface/bundle SF_01/qos/class eng3> **add_port default**

This example makes this port a default port.

applicable models:

All models.

NOTE: To configure several ports, you cannot specify a range of ports. Ports must be specified individually.

configure interface bundle qos class add_src_ip

This command assigns a source IP address or a subnet to the specified class.

parameter	definition
ip_address	Source IP address to be added to a QoS class.
netmask	The subnet mask

syntax 1:

add_src_ip ip_address < *IP address* > [netmask < *subnet mask* >]

syntax 2:

add_src_ip < default >

example 1:

Router/configure/interface/bundle SF_01/qos/class eng3> **add_src_ip 10.1.4.0 255.255.255.0**

This example adds subnet 10.1.4.0 to the class named eng3.

example 2:

Router/configure/interface/bundle SF_01/qos/class eng3> **add_src_ip default**

This example makes this class a default class.

example 3:

Router/configure/interface/bundle SF_01/qos/class eng3> **add_src_ip 10.1.4.1- 10.1.4.20**

applicable models:

All models.

configure interface bundle qos class add_vlan_id

This command assigns a VLAN identifier to the specified QoS class.

parameter	definition
vlan_id	The VLAN id number, range, or default The range is 1 - 4095, single entry (100) or a range (200 - 300).

syntax:

add_vlan_id vlan_id < n | default >

example 1:

Router/configure/interface/bundle SF_01/qos/class eng3> **add_vlan_id 100**

This example assigns vlan id "100" to the class eng3.

example 2:

Router/configure/interface/bundle SF_01/qos/class eng3> **add_vlan_id default**

This example makes the class "eng3" the default class for VLAN Ids.

related commands:

configure interface bundle qos class delete_vlan_id

applicable models:

All models.

configure interface bundle qos class burst_rate

This command modifies the maximum permitted rate.

parameter	definition
br	New burst rate for a specified QoS class (entered in Kbps) This value cannot exceed the burst rate of the parent class, and cannot be less than the committed rate for the class.

syntax:

burst_rate br < n >

example:

Router/configure/interface/bundle SF_01/qos/class eng3> **burst_rate 6000**

applicable models:

All models.

configure interface bundle qos class committed_rate

This command modifies the committed rate.

parameter	definition
cr	New committed rate for a specified QoS class (entered in Kbps) This value cannot exceed the parent committed rate.

syntax:

committed_rate cr < n >

example:

Router/configure/interface/bundle SF_01/qos/class eng3> **committed_rate 3000**

applicable models:

All models.

configure interface bundle qos class delete_dscp

This command deletes DiffServ code points from this traffic class.

A range can also be specified.

parameter	definition
ds_codepoint	Diffserv code point value The range is 0 - 63.

syntax:

delete_dscp ds_codepoint < n >

example:

Router/configure/interface/bundle qan1/qos/class v100> **delete_dscp 49**

applicable models:

All models.

configure interface bundle qos class delete_ip_address

Deletes a source or destination IP address or subnet assigned to the specified class.

parameter	definition
ip_address	Source or destination IP address or subnet to be deleted from a QoS class.
netmask	The subnet mask

syntax:

delete_ip_address ip_address < *IP address* > [netmask < *subnet mask* >]

example 1:

Router/configure/interface/bundle SF_01/qos/class eng3> **delete_ip_address 10.1.3.0**

This example deletes the IP address 10.1.3.0.

example 2:

Router/configure/interface/bundle SF_01/qos/class eng3> **delete_ip_address 10.1.3.0 255.255.255.0**

This example deletes the deletes the subnet 10.1.3.0.

example 3:

Router/configure/interface/bundle SF_01/qos/class eng3> **delete_ip_address 10.1.4.1-10.1.4.20**

applicable models:

All models.

configure interface bundle qos class delete_port

This command deletes an application port assigned to the specified class.

parameter	definition
port	The application port to be deleted from a QoS class.

syntax:

delete_port port < n | default >

example 1:

Router/configure/interface/bundle SF_01/qos/class eng3> **delete_port 80**

This example deletes port 80 from this class.

example 2:

Router/configure/interface/bundle SF_01/qos/class eng3> **delete_port default**

This example deletes the default port from this class.

applicable models:

All models.

configure interface bundle qos class delete_vlan_id

This command deletes a VLAN identifier assigned to the specified class.

parameter	definition
vlan_id	VLAN id number, range, or default The range is 1 - 4095, single entry (100) or a range (200 - 300).

syntax:

delete_vlan_id vlan_id < n | n1 - n2 | default >

example 1:

Router/configure/interface/bundle SF_01/qos/class eng3> **delete_vlan_id 100**

This example deletes VLAN id "100" from class eng3.

example 2:

Router/configure/interface/bundle SF_01/qos/class eng3> **delete_vlan_id default**

This example deletes VLAN id "default" from class eng3.

related commands:

configure interface bundle qos class add_vlan_id

applicable models:

All models.

configure interface bundle qos class enable

This command enables or disables RED or DiffServ compliant WRED.

parameter	definition
red_type	
red	Enable RED on class
ds_red	Enable DiffServ compliant WRED

syntax:

[no] enable red_type < red | ds_red >

example:

Router/configure/interface/bundle wan 1/qos/class v100> **enable red**

applicable models:

All models.

configure interface bundle qos class ewf

This command configures the exponential weight factor for the average queue size calculation.

parameter	definition
ewf	The range is 1 - 15; the default is 5.

syntax:

[no] ewf ewf < n >

example:

Router/configure/interface/bundle wan 1/qos/class v100> **ewf 10**

applicable models:

All models.

configure interface bundle qos class mark_dscp

This command marks all packets with a specific DiffServe code point.

parameter	definition
ds_codepoint	DiffServ code point value The range is 0 - 63.

syntax:

mark_dscp ds_codepoint < n >

example:

Router/configure/interface/bundle wan 1/qos/class v100> **mark_dscp 10**

applicable models:

All models.

configure interface bundle qos class nat_ip

This command specifies the NAT translation address to be used for packets matching this class. NAT should be enabled to make use of this feature.

parameter	definition
ip_address	Translation IP address

syntax:

nat_ip ip_address < ip_address >

example:

RouterE/configure/interface/bundle SF_01/qos/class eng3> **nat_ip 10.1.3.1**

applicable models:

All models.

configure interface bundle qos class mark_vlan

This command tags all packets with a specific VLAN ID when VLAN tagging is enabled.

parameter	definition
vlan_id	The range is 1 - 4095.

syntax:

mark_vlan vlan_id < n >

example:

Router/configure/interface/bundle wan 1/qos/class v100> **mark_vlan vlan_id 105**

applicable models:

All models.

configure interface bundle qos class priority

This command changes the scheduling priority for this class.

parameter	definition
priority	Scheduling priority The range is 1 - 8.

syntax:

priority priority < n >

example:

Router/configure/interface/bundle wan 1/qos/class v100> **priority 6**

applicable models:

All models.

configure interface bundle qos class queue_buffers

This command sets the maximum buffer limit for this class.

Min and max limits and the default value are indicated on the configuration display for the class.

parameter	definition
maximum	Maximum buffer limit value

syntax:

queue_buffers [maximum < n >]

example:

Router/configure/interface/bundle wan 1/qos/class v100> **queue_buffers 70**

applicable models:

All models.

configure interface bundle qos class red

This command configures Random Early Drop (RED) for the class.

parameter	definition
minth	Minimum threshold
maxth	Maximum threshold
mpd	Mark probability denominator (as an exponent of 2)
dscp	DiffServ code point (for WRED)

syntax:

[no] red [minth < n >] [maxth < n >] [mpd < n >] [dscp < n >]

example 1:

Router/configure/interface/bundle wan 1/qos/class v100> **red minth 10 maxth 30**

example 2:

Router/configure/interface/bundle wan 1/qos/class v100> **red mpd 6**

example 3:

Router/configure/interface/bundle wan 1/qos/class v100> **red minth 12 maxth 24 dscp af11**

applicable models:

All models.

configure interface bundle qos delete_all

This command deletes all QoS classes configured for the specified bundle.

syntax:

delete_all

example:

Router/configure/interface/bundle SF_01/qos> **delete_all**

applicable models:

All models.

NOTE: This command deletes all classes for a specific bundle, both inbound and outbound.

configure interface bundle qos delete_class

This command deletes a QoS class configured on the specified bundle. Only a leaf class (a class that has no children) can be deleted.

parameter	definition
class_name	Name of QoS class to be deleted from a specified bundle.

syntax:

delete_class class_name < *name* >

example:

Router/configure/interface/bundle SF_01/qos> **delete_class eng3**

applicable models:

All models.

configure interface bundle qos enable

This command enables class-based queuing on a specified interface. CBQ must be enabled for QoS classes to function.

CBQ bandwidths (CR and BR) can be configured as a percentage of the interface bandwidth, from 1 – 100%.

CR and BR Kbps for all classes on a bundle for a given direction should be expressed either as a percentage or in Kbps. Do not mix percentage and Kbps.

Policing rate can also be configured as a percentage of the interface bandwidth, however policing rates for classes on a bundle can be configured as a mix of Kbps and percentage.

The **save local** command will save bandwidths (CR, BR, policing rate) in the same way as they were configured.

If bandwidths are configured as a percentage, when the bandwidth of a multi-link bundle is modified by adding or removing links, the bandwidths will automatically be updated based on the new interface bandwidth and buffer allocation for the interface will also be updated to match the new interface bandwidth.

NOTE: When links in a bundle go down (due to an alarm for example), the class bandwidths are not updated. The percentage configuration is intended to be a percentage of the configured interface bandwidth. .Similarly class bandwidths are not updated for fast-ethernet (FE) interfaces, when speed changes between 10 Mbps and 100 Mbps in auto-negotiate mode.

FE interfaces are in auto-negotiate mode by default. On FE interfaces in this mode, during initial configuration of QoS, the desired negotiation to 100 Mbps might not have yet happened. For this reason, percentage bandwidth on FE interfaces is always calculated as a percentage of 100 Mbps. In rare situations, users might want to operate the FE interface with 10 Mbps. In this case, bandwidth must be configured in Kbps.

parameter	definition
feature	
cbq	Enable CBQ on the interface
policing	Enable policing on the interface
mon	Enable only monitoring
direction	
outbound	Outbound traffic
inbound	Inbound traffic

syntax:

enable feature < cbq | mon > direction < outbound | inbound >

example:

host/configure/interface/bundle Tokyo_6/qos> **enable cbq outbound**

applicable models:

All models.

configure interface bundle red

This command accesses next-level commands for configuring RED on bundles.

RED provides a means of avoiding traffic congestion at the Router system. Disable RED only if it might adversely affect traffic flow.

syntax:

[no] red

example:

Router/configure/interface/bundle SF_01> **red**

next-level commands

configure interface bundle red tx_max_thresh
configure interface bundle red tx_min_thresh
configure interface bundle red wq_bias_factor

applicable models:

All models.

NOTE: For frame relay bundles, RED is configured on PVCs.

configure interface bundle red tx_max_thresh

This command sets the maximum number of buffers in the bundle queue.

If this value is exceeded, RED drops all packets until the queue size drops below the selected value.

parameter	definition
value	Maximum number of buffers in the bundle's queue For better results, set the maximum threshold at least twice the value of the minimum threshold setting. The range is 2 - 511; the default is dependent upon the bandwidth of the bundle.

syntax:

tx_max_thresh value < n >

example:

Router/configure/interface/bundle SF_01/red> **tx_max_thresh 14**

applicable models:

All models.

configure interface bundle red tx_min_thresh

This command sets the minimum queue size after which RED will begin dropping packets.

If this value is exceeded during periods of traffic congestion, RED starts dropping packets with a probability that is directly proportional to the queue size.

parameter	definition
value	The threshold at which RED begins dropping packets, stated as the minimum number of buffers in the bundle's output queue. Setting the minimum threshold below 10 may cause unnecessary packet drops. The range is 1 - 511; the default is dependent upon the bandwidth of the bundle. The default setting is applied when a bundle is configured. To check the default value, issue the **show interface bundle** command.

syntax:

tx_min_thresh value < n >

example:

Router/configure/interface/bundle SF_01/red> **tx_min_thresh 10**

applicable models:

All models.

configure interface bundle red wq_bias_factor

This command sets the probability factor for dropping packets when RED is enabled.

A low value increases the probability of packet drops and reduces the bundle's queue size. An extremely low value may cause RED to drop more packets than necessary during temporary traffic congestion. A high value reduces the probability of packet drops and increases the queue size, making it better for temporary traffic bursts.

Router recommends that you do not change the **wq_bias_factor** value after setting it.

parameter	definition
value	Bias/weight factor for the RED average queue size calculation. The range is 3 - 20; the default is 5.

syntax:

wq_bias_factor value < n >

example:

Router/configure/interface/bundle SF_01/red> **wq_bias_factor 6**

applicable models:

All models.

configure interface bundle restore

This command defines how a dropped link of a multilink bundle is restored.

You can have the system do this automatically (default setting), or you can do it manually by using the **configure interface bundle link_restore** command.

parameter	definition
automatic	Automatically restores a link to a bundle if that link is free of the specified type of error for n seconds (see below).
	This is the default restoral mode.
n	Error-free time interval (seconds) that must pass before a link is automatically restored.
	The default is 120 seconds.
manual	Requires you to restore the link manually via the **configure interface bundle link_restore** command.

syntax:

restore < automatic < n > | manual >

example:

Router/configure/interface/bundle SF_01> **restore automatic 20**

applicable models:

All models.

NOTE: If **link_restore** is set to manual and a T1 or E1 link is dropped from a bundle, reconfiguring link restore to automatic does not bring link back into bundle. User should restore any dropped links manually before reconfiguring for automatic link restore.

configure interface bundle rtp

This command enables or disables RTP header compression and accesses next-level commands for configuring RTP.

Before enabling RTP, the bundle must be configured with links, protocol, and IP address and netmask.

syntax:

[no] rtp

example:

Router/configure/interface/wan1> **rtp**

next-level commands

configure interface bundle rtp connections
configure interface bundle rtp timeout

applicable models:

All models.

configure interface bundle rtp connections

This command configures the maximum number of RTP connections for the bundle.

parameter	definition
connections	The number of connections The range is 1 - 512; the default is 512.

syntax:

[no] connections connections < n >

example:

Router/configure/interface/bundle wan1/rtp> **connections 70**

related commands:

configure interface bundle rtp timeout

applicable models:

All models.

configure interface bundle rtp timeout

This command configures the timeout for the RTP table entries.

RTP timeout applies to both the compressor and decompressor entries on the same router. If no packets are sent or received during the configured timeout period, then that entry is deleted from the table.

parameter	definition
seconds	Timeout in seconds The range is 30 - 65535; the default is 60.

syntax:

timeout seconds < n >

example:

Router/configure/interface/bundle wan1/rtp> **timeout 80**

related commands:

configure interface bundle rtp connections

applicable models:

All models.

configure interface bundle shutdown

This command shuts down a WAN bundle by stopping all data transmission over it.

syntax:

[no] shutdown

example:

Router/configure/interface/bundle SF_01> **shutdown**

applicable models:

All models.

configure interface bundle src_addr

This command assigns the local IP address to be used in PPP/MLPPP negotiations.

parameter	definition
ipaddress	Configure the negotiating IP address of the bundle The default value is the source forwarding IP address of the bundle.

syntax:

src_addr [ipaddress < *IP address* >]

example:

Router/configure/interface/bundle Toronto> **src_addr 100.110.10.1**

applicable models:

All models.

configure interface bundle track

This command accesses next-level commands for configuring tracking parameters.

syntax:

track

example:

configure/interface/bundle wan1> **track**

next-level commands

configure interface bundle track hold_down
configure interface bundle track interface

applicable models:

All models.

configure interface bundle track hold_down

This command configures the hold down time for the tracking interface.

parameter	definition
down_time	Hold down interval in seconds The range is 1 - 100: the default is 5.

syntax:

track hold-down [down_time]

example:

configure/interface/bundle wan1> **track hold_down 60**

related commands:

configure interface bundle track interface

applicable models:

All models.

configure interface bundle track interface

This command configures tracking on the interface.

parameter	definition
bundle_name	Name of bundle to be tracked

syntax:

track interface bundle_name

example:

configure/interface/bundle wan1> **track interface wan1**

related commands:

configure interface bundle track hold_down

applicable models:

All models.

configure interface bundle vlan

This command accesses next-level commands to configure VLAN parameters for this interface.

syntax:

[no] vlan

example:

Router/configure/interface/bundle SF_01> **vlan**

next-level commands

configure interface bundle vlan router_ip_addr
configure interface bundle vlan vlanid

applicable models:

All models.

configure interface bundle vlan router_ip_addr

This command configures a remote router IP address for a VLAN ARP request.

syntax:

[no] router_ip_addr ip_address < *IP address* >

example:

Router/configure/interface/bundle SF_01/vlan> **router_ip_addr 150.42.3.2**

related commands:

configure interface bundle vlan vlanid

applicable models:

All models.

configure interface bundle vlan vlan_ether_type

This command configures the VLAN Ethernet type.

parameter	definition
vlan_ether_type	Ethernet type in decimal format The range is 1 - 65536; the default is 33024 (0x8100).

syntax:

[no] vlan_ether_type < n >

example:

Router/configure/interface/bundle wan1/vlan> **vlan_ether_type 243**

applicable models:

All models.

configure interface bundle vlan vlanid

This command assigns a VLAN ID for a specific bundle and enables VLAN tagging for all incoming packets on that interface.

parameter	definition
vlanid	The VLAN ID number The range is 1 - 4095.

syntax:

[no] vlanid vlanid < n >

example:

Router/configure/interface/bundle SF_01/vlan> **vlanid 200**

related commands:

configure interface bundle vlan router_ip_addr

applicable models:

All models.

configure interface bundle vlan vld_ether_type

This command configures the vld Ethernet type.

parameter	definition
vld_ether_type	Ethernet type in decimal format The range is 1 - 65536; the default is 33024 (0x8100).

syntax:

[no] vld_ether_type < n >

example:

Router/configure/interface/bundle wan1/vlan> **vld_ether_type 243**

applicable models:

All models.

configure interface bundle vlan vldid

This command configures the vld tag for this interface.

parameter	definition
vldid	The vld id number The range is 1 -4095.

syntax:

[no] vld < n >

example:

Router/configure/interface/bundle wan1/vlan> **vld 243**

applicable models:

All models.

configure interface drop_insert

This command accesses next-level commands for configuring drop and insert multiplexing functionality.

parameter	definition
intf_name	The name of the drop and insert interface. The name is limited to eight characters.

syntax:

interface drop_insert intf_name

example:

Router-OmniAccess 604/configure> **interface drop_insert Router**

next-level commands

configure interface drop_insert link
configure interface drop_insert mode

applicable models:

OmniAccess 604

configure interface drop_insert link

This command accesses next-level commands to configure links for drop and insert multiplexing.

syntax:

link

example:

Router-OmniAccess 604/configure/interface/drop_insert Router> **link**

next-level commands

configure interface drop_insert link e1
configure interface drop_insert link t1

applicable models:

OmniAccess 604

configure interface drop_insert link e1

This command configure E1 links for drop_insert multiplexing.

parameter	definition
PBX_port	The PBX port: E1, number 1
NW_port	The network port: E1, number 2
timeslots	Timeslots for drop and insert multiplexing
	The default is 1-24. Certain timeslots will be reserved depending on the signaling type configured.
signaling	Bypass signaling for drop and insert multiplexing
	The range is 1 - 2; the default is 1 (RBS).
1	RBS (Robbed Bit Signaling)
2	ISDN (Timeslot 24 is reserved for signaling.)

syntax:

e1 PBX_port < n > NW_port < n > [timeslots < n | n - n >] [signaling < 1 | 2 >]

example:

Router-OmniAccess 604E/configure/interface/drop_insert Router> **link e1 1 2 timeslots 1-10**

applicable models:

OmniAccess 604E

configure interface drop_insert link t1

This command configure T1 links for drop_insert multiplexing.

parameter	definition
PBX_port	The PBX port: T1, number 1
NW_port	The network port: T1, number 2
timeslots	Timeslots for drop and insert multiplexing
	The default is 1-24. Certain timeslots will be reserved depending on the signaling type configured.
signaling	Bypass signaling for drop and insert multiplexing
	The range is 1 - 2; the default is 1 (RBS).
1	RBS (Robbed Bit Signaling)
2	ISDN (Timeslot 24 is reserved for signaling.)

syntax:

t1 PBX_port < n > NW_port < n > [timeslots < n | n - n >] [signaling < 1 | 2 >]

example:

Router-OmniAccess 604/configure/interface/drop_insert Router> **link t1 1 2 timeslots 1-10**

applicable models:

OmniAccess 604

configure interface drop_insert mode

This command configures the operational mode for drop and insert multiplexing.

parameter	definition
NW_port	The network port: T1 or E1, number 2
mode_type	Operational mode
0	Data only
1	Data and voice
2	Data only

syntax:

mode NW_port < n > mode_type < 0 | 1 | 2 >

example:

Router-OmniAccess 604/configure/interface/drop_insert Router> **mode 2 1**

applicable models:

OmniAccess 604

configure interface ethernet

This command accesses next-level commands for configuring an Ethernet port. Each router has two Ethernet ports (0 and 1).

To view the current configuration of an Ethernet port, use the **show interface ethernet** command. To view a summary of information for both ports, use the **show interface ethernets** command.

parameter	definition
ethernet_identifier	Number of the Ethernet port to be configured, either 0 or 1.

syntax:

interface ethernet ethernet_identifier < 0 | 1 >

example:

Router/configure> **interface ethernet 0**

next-level commands

configure interface ethernet description
configure interface ethernet failover
configure interface ethernet icmp
configure interface ethernet ip

applicable models:

All models.

configure interface ethernet description

This command provides a brief description for an Ethernet port.

The descriptive string, enclosed in quotation marks, can have up to 15 characters.

parameter	definition
descr	Description Use up to 15 characters; enclose in quotation marks.

syntax:

description descr < *"string"* >

example:

Router/configure/interface/ethernet 0> **description "Main LAN"**

applicable models:

All models.

configure interface ethernet dhcp_relay

This command next-level commands for configuring dynamic host configuration protocol (DHCP) relay agent for Ethernet interfaces.

syntax:

dhcp_relay

example:

Router/configure/interface/ethernet 0> **dhcp_relay**

next-level commands

configure interface ethernet dhcp_relay gateway_address
configure interface ethernet dhcp_relay server_address

applicable models:

All models.

configure interface ethernet dhcp_relay gateway_address

This command configures the giaddr field of dhcp relayed packets.

If not specified, the giaddr field is marked with the IP address of interface.

syntax:

[no] dhcp_relay gateway_address < *gateway IP address* >

example:

Router/configure/interface ethernet 0> **dhcp_relay gateway_address 100.4.3.2**

This example configures the gateway IP address 100.4.3.2 for dhcp relayed packets on Ethernet 0.

related commands:

configure interface ethernet dhcp_relay server_address

applicable models:

All models.

configure interface ethernet dhcp_relay server_address

This command configures the dhcp server address to forward dhcp packets.
It also enables/disables the dhcp relay funtionality.

syntax:

[no] dhcp_relay server_address < *server IP address* >

example:

Router/configure/interface ethernet 0> **dhcp_relay server_address 120.5.4.3**

This example configures the server IP address 120.5.4.3 for dhcp relayed packets on Ethernet 0.

related commands:

configure interface ethernet dhcp_relay gateway_address

applicable models:

All models.

configure interface ethernet encapsulation

Enables encapsulation on the specified Ethernet interface or subinterface for VLAN traffic.

parameter	definition
protocol	
dot1q	Specifies a 802.1Q VLAN.
vlanid	Specifies the VLAN. Valid range is 1-4095.

syntax:

encapsulation protocol [vlanid] <cr>

example:

To enable encapsulation for VLAN 10, enter:

```
R2/configure/interface/ethernet 0> encapsulation dot1q 10
```

applicable models:

All models.

configure interface ethernet failover

This command enables or disables failover on an Ethernet port; enters a hold down time for restoring the Ethernet port once a failure has cleared.

parameter	definition
holddown	Enters the hold down time (in seconds) for restoring an Ethernet port once a failure has cleared. The range is 1 - 900; the default is 3.

syntax:

[no] failover [holdown < n >]

example:

Router/configure/interface/ethernet 0> **failover 5**

applicable models:

All models.

configure interface ethernet icmp

This command accesses next-level commands for allowing ICMP messages to be sent on an Ethernet port.

syntax:

icmp

example:

Router/configure/interface/ethernet 0> **icmp**

next-level commands

configure interface ethernet icmp redirect
configure interface ethernet icmp unreachable

applicable models:

All models.

configure interface ethernet icmp redirect

This command allows the Ethernet port to send ICMP redirect messages when a better route exists for a destination IP address.

syntax:

[no] redirect

example:

Router/configure/interface/ethernet 0/icmp> **redirect**

related commands:

configure interface ethernet icmp unreachable

applicable models:

All models.

configure interface ethernet icmp unreachable

This command allows the Ethernet port to send ICMP unreachable messages when no route exists for a destination IP address.

syntax:

[no] unreachable

example:

Router/configure/interface/ethernet 0/icmp> **no unreachable**

related commands:

configure interface ethernet icmp redirect

applicable models:

All models.

configure interface ethernet ip

This command accesses next-level commands for assigning an IP address and subnet mask to an Ethernet port as well as configuring IP packet filtering, directed broadcasting, and multicasting.

syntax:

ip

example:

Router/configure/interface/ethernet 0> **ip**

next-level commands

configure interface ethernet ip address
configure interface ethernet ip access-group
configure interface ethernet ip directed_broadcast
configure interface ethernet ip helper_address
configure interface ethernet ip multicast
configure interface ethernet ip proxy_arp

applicable models:

All models.

configure interface ethernet ip access-group

This command applies a packet filtering rule set to an Ethernet port.
You must first use the **configure ip access-list** commands to create the rule set.

parameter	definition
name	Name of the filtering rule set to be applied.
direction of packet transmission	
in	Packets that are inbound.
out	Packets that are outbound.

syntax:

ip access-group < *name* > direction < in | out >

example:

Router/configure/interface/ethernet 0> **ip access-group Rules_01 out**

related commands:

configure interface ethernet ip address
configure interface ethernet ip directed_broadcast
configure interface ethernet ip helper_address
configure interface ethernet ip multicast
configure interface ethernet ip proxy_arp

applicable models:

All models.

configure interface ethernet ip address

This command assigns an IP address and subnet mask to the Ethernet port.

NOTE: The network 1.1.0.0 is utilized internally in the Router system. The user is prevented from configuring IP addresses within, or IP routes to, this network.

parameter	definition
ipaddress	IP address of the Ethernet port (dotted decimal format)
netmask	Mask used to determine the subnet of the IP address.

syntax:

ip address ipaddress < *IP address* > netmask < *subnet mask* >

example:

Router/configure/interface/ethernet 0> **ip address 10.1.100.21 255.255.0.0**

related commands:

configure interface ethernet ip access-group
configure interface ethernet ip broadcast
configure interface ethernet ip helper_address
configure interface ethernet ip multicast
configure interface ethernet ip proxy_arp

applicable models:

All models.

configure interface ethernet ip directed_broadcast

This command enables or disables forwarding of directed broadcasts from this interface. The default value for this command is enabled.

syntax:

[no] ip directed_broadcast

example:

Router/configure/interface/ethernet 0> **ip directed_broadcast**

related commands:

configure interface ethernet ip address
configure interface ethernet ip access-group
configure interface ethernet ip helper_address
configure interface ethernet ip multicast
configure interface ethernet ip proxy_arp

applicable models:

All models.

configure interface ethernet ip helper_address

This command configures the ip helper on the interface.

syntax:

helper_address ip_address < ip address >

example:

Router/configure/interface/ethernet 0/ip> **helper_address 10.1.1.1**

related commands:

configure interface ethernet ip address
configure interface ethernet ip access-group
configure interface ethernet ip directed_broadcast
configure interface ethernet ip multicast
configure interface ethernet ip proxy_arp

applicable models:

All models.

configure interface ethernet ip multicast

This command sets the IP multicast mode on an Ethernet port.

parameter	definition
mode	Desired multicast mode, as follows:
pass	Pass all multicast data packets.
block	Block all multicast data packets.
ospfrip2	Pass OSPF and RIP(v2) packets.

syntax:

ip multicast mode < pass | block | ospfrip2 >

example:

Router/configure/interface/ethernet 0> **ip multicast pass**

related commands:

configure interface ethernet ip address
configure interface ethernet ip access-group
configure interface ethernet ip directed_broadcast
configure interface ethernet ip helper_address
configure interface ethernet ip proxy_arp

applicable models:

All models.

configure interface ethernet ip proxy_arp

This command enables proxy arp.

syntax:

proxy_arp

example:

Router/configure/interface/ethernet 0/ip> **proxy_arp**

related commands:

configure interface ethernet ip address
configure interface ethernet ip access-group
configure interface ethernet ip directed_broadcast
configure interface ethernet ip helper_address
configure interface ethernet ip multicast

applicable models:

All models.

configure interface ethernet mtu

This command configures the mtu for an Ethernet interface.

When a frame size larger than 1500 (jumbo frames) is configured, the system must be rebooted for the change to take effect. For a frame size smaller than 1500, rebooting is not required for the change to take effect.

Jumbo frames are Ethernet frames used to efficiently transfer bulk data. Both ends must support and be configured for jumbo frames to use them.

parameter	definition
mtusize	The configured size of the mtu The range is 64 - 4072; the default is 1500.

syntax:

mtu mtusize [< n >]

example:

Router/configure/interface/ethernet 0> **mtu 4072**

applicable models:

All models.

configure interface ethernet nat

This command accesses next-level commands for enabling NAT on an Ethernet port and accessing the associated NAT configuration commands.

Before using this command, make sure that the Ethernet is enabled.

syntax:

[no] nat

example:

Router/configure/interface/ethernet 0> **no nat**

next-level commands

configure interface ethernet nat address
configure interface ethernet nat enable
configure interface ethernet nat ip
configure interface ethernet nat ip max_entries
configure interface ethernet nat ip max_ports
configure interface ethernet nat pass_thru
configure interface ethernet nat port
configure interface ethernet nat reverse
configure interface ethernet nat timeout
configure interface ethernet nat trans_addr
configure interface ethernet nat trans_mode
configure interface ethernet nat unregistered

applicable models:

All models.

configure interface ethernet nat address

This command adds a local static IP address to the LAN port's translation table.

parameter	definition
local_address	The local IP address to be translated.
global_address	The global address to which the local address will be translated.

syntax:

[no] address local_address < *local IP address* > global_address < *global IP address* >

example:

Router/configure/interface/ethernet 0/nat> **address 10.10.1.5 150.157.99.2**

The above command entry will translate local address 10.10.1.5 on a private network to global address 150.157.99.2 on the public network (Internet).

applicable models:

All models.

configure interface ethernet nat enable

This command enables static address and port translation, and enables dynamic port translation on a bundle.

Static translation is factory-enabled on all Router systems, and dynamic translation is factory-disabled. Type **no enable static** or **no enable dynamic** to disable a previously enabled form of translation.

parameter	definition
static	Static address and port translation
dynamic	Dynamic port translation

syntax:

[no] enable < static | dynamic >

example:

Router/configure/interface/ethernet 0/nat> **enable dynamic**

applicable models:

All models.

configure interface ethernet nat ip

This command changes the factory default IP address used for dynamic port translation. The default is the Ethernet port IP address.

parameter	definition
old_ip_address	Old translation IP address
new_ip_address	New global IP address to which the LAN port's current IP address will be translated.

syntax:

ip old_ip_address < *IP address* > new_ip_address < *IP address* >

example:

Router/configure/interface/ethernet 0/nat> **ip 140.141.99.29**

applicable models:

All models.

configure interface ethernet nat max_entries

This command sets the maximum number of address translations that can occur on a LAN port.

parameter	definition
max_translations	Maximum number of translations

syntax:

max_entries max_translations < n >

example:

Router/configure/interface/ethernet 0/nat> **max_entries 1000**

applicable models:

All models.

configure interface ethernet nat pass_thru

This command enables pass_thru on an Ethernet port.

When pass_thru is enabled, all incoming and outgoing packets without translation table entries are passed through the system unchanged. This function is factory-enabled, however, you can disable it for additional security.

syntax:

[no] pass_thru

example:

Router/configure/interface/ethernet 0/nat> **no pass_thru**

applicable models:

All models.

configure interface ethernet nat port

This command adds static ports to an Ethernet port for translation. Static ports may be added for both TCP- and UDP-protocol ports.

parameter	definition
protocol_type	
tcp	TCP protocol
udp	UDP protocol
local_address	Local address
local_port	Local port of the local address
	The range is 1 - 65535.
global_address	Global address
global_port	Global port of the global address
	The range is 1 - 65535.

syntax:

port [no] protocol_type < tcp | udp > local_address < *local IP address* > local_port < n > global_address < *global IP address* > global_port < n >

example:

Router/configure/interface/ethernet 0/nat> **port tcp 100.10.32.1 42 100.141.99.30 63**

applicable models:

All models.

configure interface ethernet nat reverse

This command enables or disables reverse NAT, which occurs on incoming packets from the public network.

This translates the global addresses and ports in the Ethernet port translation table to local addresses and ports. This function is factory-disabled on all Router systems.

syntax:

[no] reverse

example:

Router/configure/interface/ethernet 0/nat> **reverse**

applicable models:

All models.

configure interface ethernet nat timeout

This command sets the NAT timeout interval for an Ethernet bundle.

Router systems are factory-configured so that all dynamic TCP translation entries that have been inactive for two hours are deleted. Also, all dynamic UDP entries inactive for one minute are deleted.

parameter	definition
protocol_type	
tcp	TCP protocol
udp	UDP protocol
seconds	Number of seconds for timeout
	The default for TCP is 7200.
	The default for UDP is 60.
	The valid range is 60 - 86400.

syntax:

timeout protocol_type < tcp | udp > [seconds < n >]

example:

Router/configure/interface/ethernet 0/nat> **timeout udp 3600**

applicable models:

All models.

configure interface ethernet nat trans_addr

This command adds or deletes the NAT translation port IP address for a specific Ethernet port.

parameter	definition
ip_address	IP address of a translation port

syntax:

[no] trans_addr ip_address < *IP address* >

example:

Router/configure/interface/ethernet 0/nat> **trans_addr 101.2.4.9**

applicable models:

All models.

configure interface ethernet nat trans_mode

This command sets the NAT translation mode to either overflow or round-robin for a specific Ethernet port.

parameter	definition
mode	
overflow	Overflow mode (default)
round_robin	Round-robin mode

syntax:

[no] trans_mode [mode < overflow | round_robin >]

example:

Router/configure/interface/ethernet 0/nat> **trans_mode round_robin**

applicable models:

All models.

configure interface ethernet nat unregistered

This command configures the system to translate only unregistered IP addresses on an Ethernet port.

All registered local addresses will pass through without packet modifications. The unregistered local addresses are those within the following ranges:

- 10.0.0.0 to 10.255.255.255
- 172.16.0.0 to 172.31.255.255
- 192.168.0.0 to 192.168.255.255

To disable this function, type **no unregistered**. If you do this, any local address (registered or unregistered) can be translated by adding the appropriate entries to the translation table or enabling dynamic port translation as previously described.

syntax:

[no] unregistered

example:

Router/configure/interface/ethernet 0/nat> **unregistered**

applicable models:

All models.

configure interface ethernet qos

This command accesses next-level Ethernet commands for adding or modifying QoS classes.

syntax:

qos

example:

Router/configure/interface/ethernet 0> **qos**

next-level commands

configure interface ethernet qos add_class
configure interface ethernet qos class
configure interface ethernet qos delete_all
configure interface ethernet qos delete_class
configure interface ethernet qos enable

applicable models:

All models.

configure interface ethernet qos add_class

This command adds a qos traffic class to the Ethernet interface.

parameter	definition
class_name	Name of the class to be added Enter a word. Use a maximum of 19 characters.
parent	Parent class. Enter a word. To add to the root, specify root-in or root-out.
cr	Committed Rate in Kbps Enter a number.
br	Burst Rate in Kbps Enter a number.
priority	Scheduling priority. The range is 1 - 8. Valid only for leaf classes.
src_ip_address	Source IP address or a range of source IP addresses
dst_ip_address	Destination IP address or a range of destination IP addresses
netmask	Subnet mask for the specified IP address
port	Application port or a range
vlan_id	VLAN ID or a range of VLAN IDs. The range is 1 - 4095.
dscp	DiffServ code point or a range The range is 0 - 63.
mark_nat_ip	Mark all packets with a specific Network Translation Address (NAT)
mark_dscp	Mark all packets with a specific DiffServ code point
mark_vlan	Tag packets with specific VLAN ID if tagging is enabled

syntax:

add_class < class_name > < parent > [cr < n >]] br < n >] [priority < n >] [src_ip_address] [dst_ip_address] [netmask] [port < n >] [vlan_id < n >] [dscp < n >] [mark_nat_ip] [mark_dscp] [mark_vlan]

example:

Router/configure/interface/ethernet 0/qos> **add_class v1100 root-out cr 1000 br 7936 priority 6**

related commands:

configure interface ethernet qos class
configure interface ethernet qos delete_all
configure interface ethernet qos delete_class
configure interface ethernet qos enable

applicable models:

All models.

configure interface ethernet qos class

This command accesses next-level commands to modify parameters of a traffic class.

syntax:

class class_name

example:

Router/configure/interface/ethernet 0/qos> **class v1200**

next-level commands

configure interface ethernet qos class add_dscp
configure interface ethernet qos class add_dst_ip
configure interface ethernet qos class add_port
configure interface ethernet qos class add_src_ip
configure interface ethernet qos class add_vlan_id
configure interface ethernet qos class delete_dscp
configure interface ethernet qos class delete_ip_address
configure interface ethernet qos class delete_port
configure interface ethernet qos class delete_vlan_id
configure interface ethernet qos class mark_dscp
configure interface ethernet qos class nat_ip
configure interface ethernet qos class mark_vlan

related commands:

configure interface ethernet qos add_class
configure interface ethernet qos delete_all
configure interface ethernet qos delete_class
configure interface ethernet qos enable

applicable models:

All models.

configure interface ethernet qos class add_dscp

This command adds diffserv code points to this class.

Assured forwarding code points can be specified by using keywords "af11 - af43." The expedited forwarding code point can be specified as "ef." Class selector code points can be specified using "cs0 - cs7."

parameter	definition
ds_codepoint	Diffserv code point or a range The range is 0 - 63.

syntax:

add_dscp ds_codepoint < n >

example:

Router/configure/interface/ethernet 0/qos/class v1100> **add_dscp 11**

related commands:

configure interface ethernet qos class add_dst_ip
configure interface ethernet qos class add_port
configure interface ethernet qos class add_src_ip
configure interface ethernet qos class add_vlan_id
configure interface ethernet qos class delete_dscp
configure interface ethernet qos class delete_ip_address
configure interface ethernet qos class delete_port
configure interface ethernet qos class delete_vlan_id
configure interface ethernet qos class mark_dscp
configure interface ethernet qos class nat_ip
configure interface ethernet qos class mark_vlan

applicable models:

All models.

configure interface ethernet qos class add_dst_ip

This command adds destination IP addresses or a destination subnet to this traffic class.

A range of IP addresses can also be specified.

NOTE: The subnet always defaults to 32 bits for ranges (255.255.255.255).

syntax:

add_dst_ip < ip_address > [netmask]

example 1:

Router/configure/interface/ethernet 0/qos/class v1100> **add dst_ip 192.168.27.0 netmask 24**

example 2:

Router/configure/interface/ethernet 0/qos/class v1100> **add dst_ip 192.168.27.1-192.168.27.20**

related commands:

configure interface ethernet qos class add_dscp
configure interface ethernet qos class add_port
configure interface ethernet qos class add_src_ip
configure interface ethernet qos class add_vlan_id
configure interface ethernet qos class delete_dscp
configure interface ethernet qos class delete_ip_address
configure interface ethernet qos class delete_port
configure interface ethernet qos class delete_vlan_id
configure interface ethernet qos class mark_dscp
configure interface ethernet qos class nat_ip
configure interface ethernet qos class mark_vlan

applicable models:

All models.

configure interface ethernet qos class add_port

This command adds application ports or a range of port numbers to this traffic class.

syntax:

add_port < n >

example:

Router/configure/interface/ethernet 0/qos/class v1100> **add_port 20-21**

related commands:

configure interface ethernet qos class add_dscp
configure interface ethernet qos class add_dst_ip
configure interface ethernet qos class add_src_ip
configure interface ethernet qos class add_vlan_id
configure interface ethernet qos class delete_dscp
configure interface ethernet qos class delete_ip_address
configure interface ethernet qos class delete_port
configure interface ethernet qos class delete_vlan_id
configure interface ethernet qos class mark_dscp
configure interface ethernet qos class nat_ip
configure interface ethernet qos class mark_vlan

applicable models:

All models.

configure interface ethernet qos class add_src_ip

This command adds a source IP address or a source subnet to this traffic class.

NOTE: The subnet always defaults to 32 bits for ranges (255.255.255.255).

syntax:

add_src_ip < ip_address > [netmask]

example 1:

Router/configure/interface/ethernet 0/qos/class v1100> **add_src_ip 192.168.27.0 netmask 24**

example 2:

Router/configure/interface/ethernet 0/qos/class v1100> **add_src_ip 192.168.27.1-192.168.27.20**

related commands:

configure interface ethernet qos class add_dscp
configure interface ethernet qos class add_dst_ip
configure interface ethernet qos class add_port
configure interface ethernet qos class add_vlan_id
configure interface ethernet qos class delete_dscp
configure interface ethernet qos class delete_ip_address
configure interface ethernet qos class delete_port
configure interface ethernet qos class delete_vlan_id
configure interface ethernet qos class mark_dscp
configure interface ethernet qos class nat_ip
configure interface ethernet qos class mark_vlan

applicable models:

All models.

configure interface ethernet qos class add_vlan_id

This command adds VLAN IDs to this traffic class.

A range of VLAN IDs can also be specified.

parameter	definition
vlan_id	VLAN ID The range is 1 - 4095. Enter a number or a range of numbers, e.g., 100-120.

syntax:

add_vlan_id vlan_id < n | range >

example:

Router/configure/interface/ethernet 0/qos/class v1100> **add_vlan_id 100-120**

related commands:

configure interface ethernet qos class add_dscp
configure interface ethernet qos class add_dst_ip
configure interface ethernet qos class add_port
configure interface ethernet qos class add_src_ip
configure interface ethernet qos class delete_dscp
configure interface ethernet qos class delete_ip_address
configure interface ethernet qos class delete_port
configure interface ethernet qos class delete_vlan_id
configure interface ethernet qos class mark_dscp
configure interface ethernet qos class nat_ip
configure interface ethernet qos class mark_vlan

applicable models:

All models.

configure interface ethernet qos class delete_dscp

This command deletes DiffServe code points from this traffic class.

A range can also be specified.

parameter	definition
ds_codepoint	DiffServe code point value The range is 0 - 63.

syntax:

delete_dscp ds_codepoint < n >

example:

Router/configure/interface/ethernet 0/qos/class v1100> **delete_dscp 49**

related commands:

configure interface ethernet qos class add_dscp
configure interface ethernet qos class add_dst_ip
configure interface ethernet qos class add_port
configure interface ethernet qos class add_src_ip
configure interface ethernet qos class add_vlan_id
configure interface ethernet qos class delete_ip_address
configure interface ethernet qos class delete_port
configure interface ethernet qos class delete_vlan_id
configure interface ethernet qos class mark_dscp
configure interface ethernet qos class nat_ip
configure interface ethernet qos class mark_vlan

applicable models:

All models.

configure interface ethernet qos class delete_ip_address

This command deletes source or destination IP addresses or subnets assigned to this traffic class.

NOTE: The subnet always defaults to 32 bits for ranges (255.255.255.255).

syntax:

delete_ip_address <ip_address > [netmask]

example:

Router/configure/interface/ethernet 0/qos/class v1100> **delete_ip_address 201.150.60.0 255.255.255.0**

related commands:

configure interface ethernet qos class add_dscp
configure interface ethernet qos class add_dst_ip
configure interface ethernet qos class add_port
configure interface ethernet qos class add_src_ip
configure interface ethernet qos class add_vlan_id
configure interface ethernet qos class delete_dscp
configure interface ethernet qos class delete_port
configure interface ethernet qos class delete_vlan_id
configure interface ethernet qos class mark_dscp
configure interface ethernet qos class nat_ip
configure interface ethernet qos class mark_vlan

applicable models:

All models.

configure interface ethernet qos class delete_port

This command deletes application ports from this traffic class.

parameter	definition
port	Application port number or range

syntax:

delete_port < n | range >

example:

Router/configure/interface/ethernet 0/qos/class v1100> **delete_port 20-21**

related commands:

configure interface ethernet qos class add_dscp
configure interface ethernet qos class add_dst_ip
configure interface ethernet qos class add_port
configure interface ethernet qos class add_src_ip
configure interface ethernet qos class add_vlan_id
configure interface ethernet qos class delete_dscp
configure interface ethernet qos class delete_ip_address
configure interface ethernet qos class delete_vlan_id
configure interface ethernet qos class mark_dscp
configure interface ethernet qos class nat_ip
configure interface ethernet qos class mark_vlan

applicable models:

All models.

configure interface ethernet qos class delete_vlan_id

This command deletes a VLAN ID or a range of VLAN IDs from this traffic class.

parameter	definition
vlan_id	VLAN ID Enter a number or a range of numbers. The range is 1 - 4095.

syntax:

delete_vlan_id < n | range >

example:

Router/configure/interface/ethernet 0/qos/class v1100> **delete_vlan_id 100-120**

related commands:

configure interface ethernet qos class add_dscp
configure interface ethernet qos class add_dst_ip
configure interface ethernet qos class add_port
configure interface ethernet qos class add_src_ip
configure interface ethernet qos class add_vlan_id
configure interface ethernet qos class delete_dscp
configure interface ethernet qos class delete_ip_address
configure interface ethernet qos class delete_port
configure interface ethernet qos class mark_dscp
configure interface ethernet qos class nat_ip
configure interface ethernet qos class mark_vlan

applicable models:

All models.

configure interface ethernet qos class mark_dscp

This command marks all packets with a specific DiffServe code point.

parameter	definition
ds_codepoint	DiffServ code point The range is 0 - 63.

syntax:

mark_dscp ds_codepoint

example 1:

Router/configure/interface/ethernet 0/qos/class v1100> **mark_dscp ef**

example 2:

Router/configure/interface/ethernet 0/qos/class v1100> **mark_dscp cs1**

related commands:

configure interface ethernet qos class add_dscp
configure interface ethernet qos class add_dst_ip
configure interface ethernet qos class add_port
configure interface ethernet qos class add_src_ip
configure interface ethernet qos class add_vlan_id
configure interface ethernet qos class delete_dscp
configure interface ethernet qos class delete_ip_address
configure interface ethernet qos class delete_port
configure interface ethernet qos class delete_vlan_id
configure interface ethernet qos class nat_ip
configure interface ethernet qos class mark_vlan

applicable models:

All models.

configure interface ethernet qos class nat_ip

This command specifies the NAT translation address to be used for packets matching this class.

NAT should be enabled to make use of this feature. NAT can be configured only for outbound classes.

syntax:

nat_ip < ip_address >

example:

Router/configure/interface/ethernet 0/qos/class v1100> **nat_ip 150.80.63.14**

related commands:

configure interface ethernet qos class add_dscp
configure interface ethernet qos class add_dst_ip
configure interface ethernet qos class add_port
configure interface ethernet qos class add_src_ip
configure interface ethernet qos class add_vlan_id
configure interface ethernet qos class delete_dscp
configure interface ethernet qos class delete_ip_address
configure interface ethernet qos class delete_port
configure interface ethernet qos class delete_vlan_id
configure interface ethernet qos class mark_dscp
configure interface ethernet qos class mark_vlan

applicable models:

All models.

configure interface ethernet qos class mark_vlan

This command tags packets with a specific VLAN ID when VLAN tagging is enabled.

parameter	definition
vlan_id	VLAN ID Enter a number or a range of numbers. The range is 1 - 4095.

syntax:

mark_vlan vlan_id < n | range >

example:

Router/configure/interface/ethernet 0/qos/class v1100> **mark_vlan 70**

related commands:

configure interface ethernet qos class add_dscp
configure interface ethernet qos class add_dst_ip
configure interface ethernet qos class add_port
configure interface ethernet qos class add_src_ip
configure interface ethernet qos class add_vlan_id
configure interface ethernet qos class delete_dscp
configure interface ethernet qos class delete_ip_address
configure interface ethernet qos class delete_port
configure interface ethernet qos class delete_vlan_id
configure interface ethernet qos class mark_dscp
configure interface ethernet qos class nat_ip

applicable models:

All models.

configure interface ethernet qos delete_all

This command deletes all QoS traffic classes on the interface.

syntax:

delete_all

example:

Router/configure/interface/ethernet 0/qos> **delete_all**

related commands:

configure interface ethernet qos add_class
configure interface ethernet qos class
configure interface ethernet qos delete_class
configure interface ethernet qos enable

applicable models:

All models.

configure interface ethernet qos delete_class

This command deletes the specified QoS traffic class.

syntax:

delete_class < class_name >

example:

Router/configure/interface/ethernet 0/qos> **delete_class v1100**

related commands:

configure interface ethernet qos add_class
configure interface ethernet qos class
configure interface ethernet qos delete_all
configure interface ethernet qos enable

applicable models:

All models.

configure interface ethernet qos enable

This command enables desired feature on this interface.

parameter	definition
feature	
qos	Enable QoS on the interface
mon	Enable monitoring only on the interface
direction	
outbound	Outbound traffic only
inbound	Inbound traffic only

syntax:

enable feature < qos | mon > direction < outbound | inbound >

example:

Router/configure/interface/ethernet 0/qos> **enable qos outbound**

related commands:

configure interface ethernet qos add_class
configure interface ethernet qos class
configure interface ethernet qos delete_all
configure interface ethernet qos delete_class

applicable models:

All models.

configure interface ethernet REM

This command allows users to add comments (at the beginning of the Ethernet area of the configuration file) during a configuration session.

Comments will appear after using the **save local** command.

parameter	definition
comments	80 character (maximum) string enclosed in quotation marks.

syntax:

REM comments < "string" >

example:

Router/configure/interface/ethernet 0> **REM "Ethernet configured on November 4, 2000.TVD 11:15."**

related commands:

display configuration stored

applicable models:

All models.

configure interface ethernet REM_

This command allows users to add comments (at the end of the Ethernet area of the configuration file) during a configuration session.

Comments will appear after using the **save local** command.

parameter	definition
comments	80-character (maximum) string enclosed in quotation marks.

syntax:

REM_ comments < "string" >

example:

Router/configure/interface/ethernet 0> **REM_ "Ethernet configured on November 4, 2000.TVD 11:15."**

related commands:

display configuration stored

applicable models:

All models.

configure interface ethernet shutdown

This command shuts down an Ethernet port.

This command disables traffic on the port and retains the port's current configuration and statistics. It also disconnects all Telnet users from the system.

To restore the port after shutting it down, log into the system from the RS-232 console port and type **no ethernet shutdown** at the configure/interface> prompt.

syntax:

shutdown

example:

Router/configure/interface/ethernet 0> **shutdown**

applicable models:

All models.

configure interface ethernet speed

Sets the speed and operating mode of an Ethernet port.

parameter	definition
speed	Ethernet speed (10, 100 Mbps, or auto to automatically adjust to the current Ethernet speed) The default is auto.
mode	Full-duplex (two-way transmission) or half-duplex (one-way transmission) The default is full-duplex. Enter this parameter only if you set **speed** to 10 or 100 Mbps.

syntax:

speed speed < 10 | 100 | auto > [mode < full_duplex | half_duplex >]

example:

Router/configure/interface/ethernet 0> **speed 100 half_duplex**

applicable models:

All models.

configure interface ethernet track

This command accesses next-level commands for configuring tracking parameters.

syntax:

track

example:

configure/interface/ethernet 0> **track**

next-level commands

configure interface ethernet track hold_down
configure interface ethernet track interface

applicable models:

All models.

configure interface ethernet track hold_down

This command configures the hold down time for the tracking interface.

parameter	definition
down_time	Hold down interval in seconds The range is 1 - 100; the default is 5.

syntax:

track hold_down [down_time]

example:

configure/interface/ethernet 0> **track hold_down 60**

related commands:

configure interface ethernet track interface

applicable models:

All models.

configure interface ethernet track interface

This command configures tracking on the interface.

parameter	definition
bundle_name	Name of bundle to be tracked

syntax:

track interface bundle_name

example:

configure/interface/ethernet 0> **track interface wan1**

related commands:

configure interface ethernet track hold_down

applicable models:

All models.

configure interface ethernet vlan

This command accesses next-level commands for configuring VLAN.

syntax:

vlan

example:

Router/configure/interface/ethernet 0> **vlan**

next-level commands

configure interface ethernet vlan vlanid

applicable models:

All models.

configure interface ethernet vlan vlan_ether_type

This command configures the VLAN Ethernet type.

parameter	definition
vlan_ether_type	Ethernet type in decimal form The range is 1 - 65535; the default is 33024 (0x8100).

syntax:

[no] vlan_ether_type < n | n - n >

example:

Router/configure/interface/ethernet 1/vlan> **vlan_ether_type 1172**

applicable models:

All models.

configure interface ethernet vlan vlanid

This command assigns a VLAN ID for a specific Ethernet port and enables VLAN tagging for all incoming packets on that interface.

parameter	definition
vlanid	The VLAN ID number The range is 1 - 4095.

syntax:

[no] vlanid vlanid < n >

example:

Router/configure/interface/ethernet 0> **vlanid 200**

applicable models:

All models.

configure interface ethernet vlan vld_ether_type

This command configures the Ethernet type in decimal format.

parameter	definition
vld_ether_type	Ethernet type in decimal form The range is 1 - 65535; the default is 33024 (0x8100).

syntax:

[no] vld_ether_type < n | n - n >

example:

Router/configure/interface/ethernet 1/vlan> **vlan_ether_type 1172**

applicable models:

All models.

configure interface ethernet vlan vldid

This command configures the vld tag for this interface.

parameter	definition
vldid	Vld id number The range is 1 - 4095.

syntax:

[no] vldid < n >

example:

Router/configure/interface/ethernet 1/vlan> **vldid 200**

applicable models:

All models.

configure interface ethernet vrrp

This command configures a VRRP group for an Ethernet interface.

parameter	definition
group	Group number The range is 1 - 255.

syntax:

vrrp group < n >

example:

Router/configure/interface/ethernet 0> **vrrp 10**

next-level commands

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All models.

configure interface ethernet vrrp advertisement_interval

This command configures the time interval for VRRP advertisements in seconds.

parameter	definition
adv_interval	Advertisement interval in seconds The range is 1 - 255; the default is 1.

syntax:

advertisement_interval adv_interval < n >

example:

Router/configure/interface/ethernet 0/vrrp 10> **advertisement_interval 360**

related commands:

configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All models.

configure interface ethernet vrrp authentication

This command configures the VRRP authentication information.

Once configured, all outgoing VRRP packets will have this authentication information and all packets received will be authenticated using this information.

parameter	definition
auth_string	Authentication string Enter a word (maximum of eight characters).

syntax:

authentication auth_string < auth_string >

example:

Router/configure/interface/ethernet 0/vrrp 10> **authentication alfxitz**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All models.

configure interface ethernet vrrp description

This command assigns a description to the VRRP group.

parameter	definition
desc_string	Description string describing group Enter a string up to 80 characters within quotation marks.

syntax:

description < "desc_string" >

example:

Router/configure/interface/ethernet 0/vrrp 10> **description "virtual router for wan"**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All models.

configure interface ethernet vrrp enable

This command enables a VRRP group.

syntax:

enable

example:

Router/configure/interface/ethernet 0/vrrp 10> **enable**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All models.

configure interface ethernet vrrp ipaddr

This command configures VRRP group virtual IP addresses.

syntax:

ipaddr < ip address >

example:

Router/configure/interface/ethernet 0/vrrp 10> **ipaddr 128.44.10.24**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All models.

configure interface ethernet vrrp learn_adv_internal

This command configures the backup router to learn the advertisement interval from the master.

syntax:

learn_adv_interval

example:

Router/configure/interface/ethernet 0/vrrp 10> **learn_adv_interval**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All models.

configure interface ethernet vrrp preempt

This command configures the virtual router to prempt the current VRRP master if it has a higher priority than the current master.

syntax:

preempt

example:

Router/configure/interface/ethernet 0/vrrp 10> **preempt**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All models.

configure interface ethernet vrrp priority

This command configures the priority level of the router within a VRRP group.

parameter	definition
level	Priority level The range is 1 - 254.

syntax:

priority level < n >

example:

Router/configure/interface/ethernet 0/vrrp 10> **priority 100**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp track

applicable models:

All models.

configure interface ethernet vrrp track

This command configures tracked interface and track priority.

parameter	definition
intfname	Interface name (e.g., ethernet0, ethernet1, or bundle name)
track_priority	Track priority The range is 1 - 254.

syntax:

track intfname track_priority < n >

example:

Router/configure/interface/ethernet 0/vrrp 10> **track ethernet1 140**

NOTE: An Ethernet interface cannot track itself. You must specify a different Ethernet interface to track. (See example above.)

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority

applicable models:

All models.

configure interface ethernet vrrp_mode

This command configures VRRP mode.

parameter	definition
mode	vrrp mode
0	VRRP mode 0 (Gratuitous ARP for an interface) This is the default.
1	VRRP mode 1 (Active/Standby)
2	VRRP mode 2 (Promiscuous mode)

syntax:

vrrp_mode mode < n >

example:

Router/configure/interface/ethernet 0/vrrp 10> **vrrp_mode 1**

applicable models:

All models.

configure interface loopback

This command configures a software interface.

Router systems support the creation of a maximum of 128 loopback interfaces.

parameter	definition
loopback_name	Name of the interface Enter a maximum of eight characters.

syntax:

loopback loopback_name < *name* >

example 1:

Local router command sequence:

Router/configure/interface> **loopback hilow**
Router/configure/interlace/loopback hilow> ip address 10.1.1.1 255.255.255.0

example 2:

Remote BGP router command sequence:

Router/configure/router> **bgp 100**
Router/configure/router/bgp 100> **neighbor 150.30.30.1 200**
Router/configure/router/bgp 100/neighbor 150.30.30.1 200> **ebgp_multihop**
Router/configure/router/bgp 100/neighbor 150.30.30.1 200> **update_source**

In this example, the remote router is connected to the peer with the address of 150.30.30.1, which will never go down. This ensures that the BGP connection will stay up.

applicable models:

All models.

configure interface loopback ip address

This command assigns an IP address and subnet mask for a loopback interface.

syntax:

ip address ipaddress < *IP address* > netmask < subnet mask >

example:

Router/configure/interface/loopback hilow> **ip address 100.4.3.2 255.255.255.0**

applicable models:

All models.

configure interface null

This command configures a null interface.

syntax:

interface null

example:

Router_ Router/configure> i**nterface null**

applicable models:

All models.

configure interface null ip unreachables

This command enables sending ICMP unreachable messages.

parameter	definition
unreachables	Enables sending ICMP unreachable messages

syntax:

ip unreachables

example:

Router/configure/interface/null> **ip unreachables**

applicable models:

All models.

configure ip

This command accesses next-level commands for configuring IP parameters.

To view the IP parameters before or after configuring them, use the appropriate **show ip** commands.

syntax:

ip

example:

Router/configure> **ip**

next-level commands

configure ip access-group
configure ip domain_name
configure ip dos
configure ip access-list
configure ip host_add
configure ip load_balance
configure ip name_server
configure ip pname_server
configure ip route
configure ip routing

applicable models:

All models.

configure ip <ipaddr> <netmask>

This command configures the IP address and subnet mask for the router.

syntax:

ip <ipaddr> <netmask>

example:

Router/configure> **ip10.1.1.10 255.255.255.240**

next-level commands

configure ip

applicable models:

All models.

configure ip access-group

This command applies a packet filtering rule set to a bundle or Ethernet.

You must first use the **configure ip access-list** commands to create the rule set.

parameter	definition
interface	
bundle name	Bundle name
ethernet0	Ethernet 0
ethernet1	Ethernet 1
direction of packet transmission	
in	Filters packets that are inbound.
out	Filters packets that are outbound.

syntax:

apply_filter interface < *bundle name* | ethernet0 | ethernet1 > direction < in | out >

example:

Router/configure/ip> **apply_filter ethernet0 in**

applicable models:

All models.

configure ip access-list

This command accesses commands for creating and modifying IP packet filtering rule sets.

After creating a rule set, use the associated commands that follow to:

- **add** rules to an existing rule set
- **insert** a new rule set
- **show** a rule set
- **delete** a rule set

These commands are described in the following command summaries.

After creating a new rule set, you can apply it to the WAN bundles and Ethernet ports of the system, using the **configure ip access-group**, **configure interface bundle ip access-group**, or **configure interface ethernet ip access-group** commands.

syntax:

[no] < *list name* >

example:

Router/configure/ip/filter_list> **Filter02**

next-level commands

configure ip access-list add
configure ip filterlist delete
configure ip filterlist insert
configure ip filterlist show

applicable models:

All models.

configure ip access-list add

This command adds a new rule to an IP packet filtering rule set.

Each rule is added with an identifying line number for future editing or deletion.

parameter	definition
rul_action	
permit	Allows access if conditions are matched.
deny	Discards access if conditions are matched.
reject	Discards a packet, and sends an ICMP host unreachable message.
protocol	Name or number of an Internet protocol. This can be one of the key words (TCP, UDP, ICMP, or IP), or an integer int he range 0 - 255 representing an IP protocol number. To match any Internet protocol, including ICMP, TCP, or UDP, use the keyword IP.
source	Source host or network address. Or, type **any** to specify a source address/wildcard of 0.0.0.0/255.255.255.255 or 0.0.0.0/32.
/wildcard	Optional wildcard bits to be applied to the source address or destination address. This entry can be an IP address (as in Cisco notation) or the number of bits (as in NetBlazer notation).
destination	Destination host or network address. Or, type **any** to specify a destination address/wildcard of 0.0.0.0/255.255.255.255 or 0.0.0.0/32.
sport	Optional entry for TCP and UDP protocols; allows the source port to be used for packet filtering. Use comparison symbols to specify ports as follows:
=p	Port number **p**, where n is 0 - 65535.
!=p	Excludes port **p**.
>p	Any port number greater than **p**
>=p	Any port number greater than or equal to **p**
<p	Any port number less than **p**
<=p	Any port number less than or equal to **p**
p1-p2	Any port number within the range **p1 - p2**
dport	Optional entry for TCP and UDP protocols; allows the destination port to be filtered. Use the same comparison symbols and port numbering described above for the source port (sport).
precedence	Optional numeric entry; allows IP header precedence field to be filtered. The range is 0 - 7.
flags	Allows TCP flags to be filtered (optional).You can specify multiple TCP flags by separating the above keywords with commas (no spaces allowed). This entry may be any of the following words:
established	Used to match an established connection (Cisco-compatible).
fin	Matches the TCP FIN header flag.
syn	Matches the TCP SYN header flag.
ack	Matches the TCP ACK header flag.
psh	Matches the TCP PSH header flag.

parameter	definition
rst	Matches the TCP RST header flag.
urg	Matches the TCP URG header flag.
icmptype	Optional numeric entry for ICMP protocol, allowing the ICMP message type to be filtered (optional, range is 0 - 255).
icmpcode	Optional numeric entry for ICMP protocol, allowing the ICMP message code to be filtered, if specified along with a message type. The range is 0 - 255.
tos	Type of service (optional numeric entry for UDP and ICMP protocols). Allows the IP header TOS field to be filtered. The range is 0 - 15.
log	Allows a logging message to be reported to the user when a rule match occurs (optional).
on	Log the matching packet on.
off	Log the matching packet off (default)

Syntax for TCP

[no] add rul_action < permit | deny > protocol < tcp > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [sport < 0 - 65535 >] [dport < 0 - 65535 >] [precedence < 0 - 7 >] [tos < 0 - 15 >] [flags < established | fin | syn | ack | psh | rst | urg >] [log < on | off >]

Syntax for UDP

[no] add rul_action < permit | deny > protocol < udp > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [sport < 0 - 65535 >] [dport < 0 - 65535 >] [precedence < 0 - 7 >] [tos < 0 - 15 >] [log < on | off >]

Syntax for ICMP

[no] add rul_action < permit | deny | reject > protocol < icmp > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [icmptype < 0 - 255 >] [icmpcode < 0 - 255 >] [precedence < 0 - 7 >] [tos < 0 - 15 >] [log < on | off >]

Syntax for IP

[no] add rul_action < permit | deny > protocol < ip > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [precedence < 0 - 7 >] [tos < 0 - 15 >] [log < on | off >]

example:

a Router/configure/ip/filter_list Rules_01> **add permit tcp 10.1.3.9/12 10.1.100.46/255.255.255.0 dport =23 flags established log on**

b Router/configure/ip/filter_list Rules_02> **add deny icmp 10.1.20.9/32 any icmptype 8 icmpcode 0 log on**

c Router/configure/ip/filter_list Rules_03> **add permit udp 10.1.10.7 any sport =12 precedence 3 tos 2**

d Router/configure/ip/filter_list Rules_04> **add deny ip 10.1.100.2 255.255.255.0 precedence 5 tos 1 log on**

The above command entries add four filtering rules to a filtering rule set (one each for the TCP, UDP, ICMP, and IP protocols).

related commands:

configure ip access-list delete
configure ip access-list insert

applicable models:

All models.

configure ip access-list delete

This command deletes a rule from an IP packet filtering rule set.

You must know the line number of the rule to delete it. To display the rules currently stored in a rule set before or after deleting them, use the **show ip access-list filter_list** command. When you delete a rule, the line numbers of successive rules in the list are revised accordingly.

parameter	definition
rul_lineno	The line number of a specific rule set The range is 1 - 65535.

syntax:

delete rul_lineno < n >

example:

Router/configure/ip/filter_list Rules_01> **delete 3**

related commands:

configure ip access-list add
configure ip access-list insert

applicable models:

All models.

configure ip access-list insert

This command inserts a new rule into an existing filtering rule set.

This command lets you enter a rule in the middle of an existing set of rules by specifying a line number location.

Before using this command, you must display the contents of the current filtering rule set using the **configure ip access-list show** command. Each rule has a unique line number for identification purposes.

parameter	definition
rule_lineno	Line number The range is 1 - 65535.
rul_action	
permit	Allows access if conditions are matched.
deny	Discards access if conditions are matched.
reject	Discards a packet, and sends an ICMP host unreachable message.
protocol	Name or number of an Internet protocol. This can be one of the key words (TCP, UDP, ICMP, or IP), or an integer int he range 0 - 255 representing an IP protocol number. To match any Internet protocol, including ICMP, TCP, or UDP, use the keyword IP.
source	Source host or network address. Or, type **any** to specify a source address/wildcard of 0.0.0.0/255.255.255.255 or 0.0.0.0/32.
/wildcard	Optional wildcard bits to be applied to the source address or destination address. This entry can be an IP address (as in Cisco notation) or the number of bits (as in NetBlazer notation).
destination	Destination host or network address. Or, type **any** to specify a destination address/wildcard of 0.0.0.0/255.255.255.255 or 0.0.0.0/32.
sport	Optional entry for TCP and UDP protocols; allows the source port to be used for packet filtering. Use comparison symbols to specify ports as follows:
=p	Port number **p**, where n is 0 - 65535.
!=p	Excludes port **p**.
>p	Any port number greater than **p**
>=p	Any port number greater than or equal to **p**
<p	Any port number less than **p**
<=p	Any port number less than or equal to **p**
p1-p2	Any port number within the range **p1** - **p2**
dport	Optional entry for TCP and UDP protocols; allows the destination port to be filtered. Use the same comparison symbols and port numbering described above for the source port (sport).
precedence	Optional numeric entry; allows IP header precedence field to be filtered. The range is 0 - 7.
flags	Allows TCP flags to be filtered (optional).You can specify multiple TCP flags by separating the above keywords with commas (no spaces allowed). This entry may be any of the following words:
established	Used to match an established connection (Cisco-compatible).
fin	Matches the TCP FIN header flag.
syn	Matches the TCP SYN header flag.
ack	Matches the TCP ACK header flag.

psh	Matches the TCP PSH header flag.
rst	Matches the TCP RST header flag.
urg	Matches the TCP URG header flag.
icmptype	Optional numeric entry for ICMP protocol, allowing the ICMP message type to be filtered (optional, range is 0 - 255).
icmpcode	Optional numeric entry for ICMP protocol, allowing the ICMP message code to be filtered, if specified along with a message type. The range is 0 - 255.
tos	Type of service (optional numeric entry for UDP and ICMP protocols). Allows the IP header TOS field to be filtered. The range is 0 - 15.
log	Allows a logging message to be reported to the user when a rule match occurs (optional).
on	Log the matching packet on.
off	Log the matching packet off (default)

syntax:

Syntax for TCP

[no] insert rule_lineno < 1 - 65535 > rul_action < permit | deny >protocol < tcp > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [sport < 0 - 65535 >] [dport < 0 - 65535 >] [precedence < 0 - 7 >] [tos < 0 - 15 >] [flags < established | fin | syn | ack | psh | rst | urg >] [log < on | off >]

Syntax for UDP

[no] insert rule_lineno < 1 - 65535 > rul_action < permit | deny > protocol < udp > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [sport < 0 - 65535 >] [dport < 0 - 65535 >] [precedence < 0 - 7 >] [tos < 0 - 15 >] [log < on | off >]

Syntax for ICMP

[no] insert rule_lineno < 1 - 65535 > rul_action < permit | deny | reject > protocol < icmp > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [icmptype < 0 - 255 >] [icmpcode < 0 - 255 >] [precedence < 0 - 7 >] [tos < 0 - 15 >] [log < on | off >]

Syntax for IP

[no] insert rule_lineno < 1 - 65535 > rul_action < permit | deny > protocol < ip > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [precedence < 0 - 7 >] [tos < 0 - 15 >] [log < on | off >]

example:

Router/configure/ip/filter_list Rules_01> **insert 4 permit tcp 10.1.10.7 any dport 50-55 precedence 4 tos 6**

The example above inserts a new rule, 4, behind existing rule 3 in the rule set. It also increments the line numbers of all successive rules in that set.

related commands:

configure ip access-list add
configure ip access-list delete

applicable models:

All models.

configure ip dhcps

This command configures the DHCP server. Use the **configure ip dhcps enable** command to enable the DHCP server, and the **configure ip dhcps interface** command to enable the DHCP server on the specified interface. User the **configure ip dhcps pool** command to configure an address pool for the DHCP server. Use the **configure ip dhcps relay** to configure the addresses of relay agents. Use the **configure ip dhcps remote_database** command to configure a remote DHCP server database.

parameter	definition
icmptype	enable -- enable the dhcp server interface -- Enable DHCP Server on interface pool -- configure dhcps address pool pool <poolname> pool <poolname> clientid <id> pool <pool name> default_router <ipaddress> pool <poolname> dnsserver <ipaddr> pool <poolname> tftpserver <ipaddr> pool <poolname> netbios-name-server <ipaddr> pool <poolname> domain <name> pool <poolname> host <ipaddress> pool <poolname> lease <time> pool <poolname> hwaddr <addr> pool <poolname> network <net> <mask> pool <poolname> exclude-range <start-ip> <end-ip> relay -- configure ip address and network of relay agents that we will listen to remote_database -- configure dhcps remote database

applicable models:

All models.

configure ip domain_name

This command assigns a domain name to the Router system.

To view the IP domain name, use the **show ip dns** command.

syntax:

domain_name < *name* >

example:

Router/configure/ip> **domain_name abcnetworks**

applicable models:

All models.

configure ip dos

This command accesses commands for configuring the denial of service (DoS) protection for both Ethernet ports 0 and 1.

syntax:

dos

example:

Router/configure/ip> **dos**

next-level commands

configure ip dos drop
configure ip dos enable

applicable models:

All models.

configure ip dos drop

This command drops all ICMP or PING packets that are destined for the Router system.

parameter	definition
icmptypes	
icmp	Drops all inbound ICMP packets.
ping	Drops all inbound PING packets.

syntax:

drop icmptypes < icmp | ping >

example:

Router/configure/ip/dos> **drop ping**

applicable models:

All models.

configure ip dos enable

This command enables or disables DoS on Ethernet 0 and Ethernet 1.

parameter	definition
ethernet0	Designates Ethernet 0.
ethernet1	Designates Ethernet 1.

syntax:

[no] enable < ethernet0 | ethernet1 >

example:

Router/configure/ip/dos> **enable ethernet0**

applicable models:

All models.

configure ip host_add

This command adds a host name and its corresponding IP address to the hosts table.

This command checks the host table and maps a host name to an IP address before making a DNS query.

parameter	definition
hostname	Name of the new host Use up to 255 characters.
hostaddress	IP address of the new host.

syntax:

host_add hostname < *host name* > hostaddress < *IP address* >

example:

Router/configure/ip> **host_add mainsys 10.1.3.251**

applicable models:

All models.

configure ip load_balance

This command configures the load balancing policy for equal cost routes between WAN bundles.

Users can choose between a per-packet or per-flow policy. If no policy is chosen, then the system will default to failover mode.

parameter	definition
policy	
per_flow	Chooses between WAN bundles based on the destination IP address.
per_packet	Alternates between bundles on a per-packet basis.

syntax:

[no] load_balance policy < per_flow | per_packet >

example:

Router/configure/ip> **load_balance per_flow**

applicable models:

All models.

configure ip name_server

This command configures a name server for the purpose of sending DNS requests.

You can configure a maximum of three name servers; including a primary server, which is the first configured server.

parameter	definition
ipaddress	IP address of the host name server added to the list of name servers.

syntax:

name_server ipaddress < IP address >

example:

Router/configure/ip> **name_server 132.10.1.5**

applicable models:

All models.

configure ip nat

This command accesses next-level commands for configuring global NAT.

syntax:

[no] nat

example:

Router/configure/ip> **nat**

next-level commands

configure ip nat address
configure ip nat default_addr
configure ip nat enable
configure ip nat interface
configure ip nat ip
configure ip nat max_entries
configure ip nat max_ports
configure ip nat pass_thru
configure ip nat pool
configure ip nat port
configure ip nat reverse
configure ip nat timeout
configure ip nat trans_addr
configure ip nat trans_mode
configure ip nat unregistered

applicable models:

All models.

configure ip nat address

This command adds a static address translation entry to the network address translation table.

parameter	definition
local_address	Local IP address to translate
global_address	Global IP address to use for the translation

syntax:

[no] address < local_address > < global_address >

example:

Router/configure/ip/nat> **address 10.10.10.1 140.55.23.41**

applicable models:

All models.

configure ip nat default_addr

This command adds or deletes a dynamic port translation IP address.

syntax:

[no] default_addr ip_address

example:

Router/configure/ip/nat> **default_addr 10.10.10.3**

NOTE: This command should be used in conjunction with the **configure ip nat enable** command.

applicable models:

All models.

configure ip nat enable

This command enables or disables network address translation modes.

parameter	definition
translationmode	
static	Specify for static address/port translation
dynamic	Specify for dynamic port translation
address	Specify for dynamic address translation
pool_name	Name of the address pool This optional parameter is only required when the address parameter is used.

syntax:

[no] enable translationmode < static | dynamic | address > [pool_name]

NOTE: After enabling NAT, execute the **configure ip nat default_addr** command.

example 1:

Router/configure/ip/nat> **enable static**

example 2:

Router/configure/ip/nat> **enable address pool1**

applicable models:

All models.

configure ip nat interface

This command adds or deletes global NAT for an interface.

parameter	definition
interface	
ethernet0	Ethernet 0 interface
ethernet1	Ethernet 1 interface
bundle_name	Enter the name of a bundle.
bundle_name:pvc number	Enter the name of a bundle and a pvc number. (For frame relay bundle use.)

syntax:

[no] interface interface < ethernet0 | ethernet1 | bundle_name | bundle_name:pvc number >

example:

Router/configure/ip/nat> **interface ethernet0**

applicable models:

All models.

configure ip nat ip

This command changes a configured dynamic port translation IP address.

parameter	definition
old_ip_address	Old or existing translation IP address
new_ip_address	New translation IP address to use

syntax:

ip old_ip_address new_ip_address

example:

Router/configure/ip/nat> **ip 100.10.10.14 100.100.10.1**

applicable models:

All models.

configure ip nat max_entries

This command limits the number of translations that can occur.

parameter	definition
max_translations	Specifies the maximum number of translations to support.

syntax:

max_entries max_translations

example:

Router/configure/ip/nat> **max_entries 44**

applicable models:

All models.

configure ip nat max_ports

This command configures the maximum ports or translations for the specified translation address.

parameter	definition
max_ports	Specifies the maximum number of ports and translations for the translated address.

syntax:

max_ports ipaddress max_translations

example:

Router/configure/ip/nat> **max_ports 143.55.34.4 140**

applicable models:

All models.

configure ip nat pass_thru

This command allows not-translated packets to pass through to the router.

syntax:

pass_thru

example:

Router/configure/ip/nat> **pass_thru**

applicable models:

All models.

configure ip nat pass-thru-multicast

This command allows multicast packets to pass through to the router.

syntax:

pass-thru-multicast

example:

Router/configure/ip/nat> **pass-thru-multicast**

applicable models:

All models.

configure ip nat pool

This command names the address pool and accesses next-level commands to configure the address pool for dynamic address translation.

syntax:

pool < name >

example:

Router/configure/ip/nat> **pool custpool1**

next-level commands

configure ip nat pool range

applicable models:

All models.

configure ip nat pool range

This command adds or deletes a range of addresses from the translation pool.

parameter	definition
start_ip	Starting IP address of the range
end_ip	Ending IP address of the range
netmask	The netmask for the specified range.

syntax:

range < start_ip > < end_ip > netmask

example:

Router/configure/ip/nat/pool custpool1> **range 23.226.58.1 23.226.58.250 24**

NOTE: The CLI does not perform any checks for overlapping ranges.

applicable models:

All models.

configure ip nat port

This command adds static a port to the network address translation table.

parameter	definition
protocol_type	
tcp	Specifies TCP protocol translation entries for the port
udp	Specifies UDP protocol translation entries for the port
addr	Specifies dynamic addressing translation entries for the port
local_address	The local IP address to translate.
local_port	The local port to translate. The range is 1 - 65535.
global_address	The global IP address to use for translation.
global_port	The global port to use for translation The range is 1 - 65535.

syntax:

port protocol_type < tcp | udp | addr > local_address < local IP address > local port < n > global address < global IP address > global_port < n >

example:

Router/configure/ip/nat> **port addr 10.10.10.1 5 142 66.32.12 11**

applicable models:

All models.

configure ip nat reverse

This command enables or disables reverse NAT functionality.

syntax:

[no] reverse

example:

Router/configure/ip/nat> **reverse**

applicable models:

All models.

configure ip nat timeout

This command configures a timeout value for dynamic translation entries.

parameter	definition
option_type	
tcp	Assigns TCP protocol translation entries to the configured timeout value
udp	Assigns UDP protocol translation entries to the configured timeout value
addr	Assigns addressing to the configured timeout value
seconds	Configures the number of seconds for the timeout The range is 60 - 86400. The defaults are: tcp: 7200 udp: 60 addr: 3600

syntax:

timeout option_type < tcp | udp | addr > [seconds]

example:

Router/configure/ip/nat> **timeout udp 300**

applicable models:

All models.

configure ip nat trans_addr

This command adds or deletes a specified dynamic port translation IP address.

syntax:

[no] trans_addr ip_address

example:

Router/configure/ip/nat> **trans_addr 10.10.10.5**

applicable models:

All models.

configure ip nat trans_mode

This command sets the translation mode to either OVERFLOW or ROUND_ROBIN.

parameter	definition
tmode	
overflow	Uses the next available translation address only if the current translation address ports have been used.
round_robin	Uses the translation ports from each of the translation addresses in a round robin fashion.

syntax:

trans_mode [mode < overflow | round_robin >]

example:

Router/configure/ip/nat> **trans_mode overflow**

applicable models:

All models.

configure ip nat unregistered

This command configures only unregistered local addresses to get translated.

syntax:

[no] unregistered

example:

Router/configure/ip/nat> **unregistered**

applicable models:

All models.

configure ip pname_server

This command defines the IP address of the primary DNS server.

parameter	definition
ipaddress	IP address of the primary name server.

syntax:

pname_server ipaddress < *IP address* >

example:

Router/configure/ip> **pname_server 10.1.100.16**

applicable models:

All models.

configure ip route

This command adds a static IP route to a Router system.

NOTE: The network 1.1.0.0 is utilized internally in the Router system. The user is prevented from configuring IP addresses within, or IP routes to, this network.

parameter	definition
ipaddress	IP address of the static route
netmask	Subnet mask of the static route
gateway	Gateway IP address, interface name, or fr bundlename:pvc# to the destination
metric	Administrative distance of the route The range is 1 - 255.

syntax:

route ipaddress < *IP address* > netmask < *subnet mask* > gateway < *gateway IP address* > metric < n >

example 1:

Gateway route:

Router/configure/ip> **route 10.1.200.0 255.255.0.0 10.2.71.5 2**

example 2:

Interface route:

Router/configure/ip> **route 10.1.200.0 255.255.255.0 wan1 2**

example 3:

Frame relay/pvc route:

Router/configure/ip> **route 10.1.200.0 255.255.255.0 wan1:17 2**

applicable models:

All models.

configure ip routing

This command enables IP routing and disables IP mux proxy operation.

When changing between IP multiplexing and IP dynamic routing modes, you must reboot the system. Then review the system configuration (in the new mode) to verify that all IP configurations and IP routes are still configured.

Use the no form of this command to disable routing and switch to IP mux mode.

syntax:

[no] routing

example:

Router/configure/ip> **routing**

applicable models:

All models.

configure ipmux

This command accesses next-level commands for configuring IP multiplexing on a Router system.

When changing between IP multiplexing and IP dynamic routing modes, you must reboot the system. Then review the system configuration (in the new mode) to verify that all IP configurations and IP routes are still configured.

syntax:

ipmux

example:

Router/configure> **ipmux**

next-level commands

configure ipmux autoconf
configure ipmux route
configure ipmux src_fwd_gw

applicable models:

All models.

configure ipmux autoconf

This command enables or disables IP mux auto-configuration on a Router system.

syntax:

autoconf

example:

Router/configure> **autoconf**

applicable models:

All models.

configure ipmux route

This command adds a destination IP address and subnet as well as the gateway IP address to the system routing table.

parameter	definition
ipaddress	IP address of the route's destination
netmask	Destination's subnet mask address
gwipaddr	IP address of the gateway to the destination
pvc	PVC number for route using fr gateway name The range is 16 - 1022; the default is 0.

syntax:

ipmux route ipaddress < *IP address* > netmask < *subnet mask* > gwipaddr < *gateway IP address* >
[pvc]

Router/configure> **ipmux route 10.2.100.16 255.0.0.0 10.2.100.30**

applicable models:

All models.

configure ipmux src_fwd_gw

This command selects the default gateway for all configured bundles.

This command avoids having to configure a source forwarding statement within each bundle.

parameter	definition
gateway	IP address

syntax:

src_fwd_gw gateway < *IP address* >

example:

Router/configure/ipmux> **src_fwd_gw 100.10.10.4**

applicable models:

All models.

configure module

This command accesses next-level commands for configuring interface ports.

syntax:

module

example:

Router/configure> **module**

next-level commands

configure module ct3
configure module e1
configure module t1
configure module t3
configure module ussi

applicable models:

All models.

configure module e1

This command accesses next-level commands for selecting ann E1 link for configuration.

To select multiple contiguous E1 links for configuration, type **e1 n-n** (where **n-n** is the range of E1s to be configured).

parameter	definition
e1_no	Number of the E1 WAN link to be configured. The range is 1 - 16, depending on the system model.

syntax:

module e1 e1_no < n >

example:

RouterE/configure> **module e1 2**

next-level commands

configure module e1 alarms
configure module e1 circuitID
configure module e1 clock_source
configure module e1 contactInfo
configure module e1 description
configure module e1 enable
configure module e1 framing
configure module e1 jitter

applicable models:

OmniAccess 601, OmniAccess 602E

configure module e1 alarms

This command accesses next-level commands for configuring alarm thresholds. on an E1 link.

syntax:

alarms

example:

RouterE/configure/module/e1 1-3> **alarms**

next-level commands

configure module e1 alarms thresholds

applicable models:

OmniAccess 601, OmniAccess 602E

configure module e1 alarms thresholds

This command accesses next-level commands for configuring alarm thresholds.

If a user-defined threshold is exceeded, the system reports an alarm. These alarms indicate the possible deterioration of an associated E1 link.

syntax:

thresholds

example:

RouterE/configure/module/e1 1-3/alarms> **thresholds**

next-level commands

configure module e1 alarms thresholds user

applicable models:

OmniAccess 601, OmniAccess 602E

configure module e1 alarms thresholds user

This command sets the E1 user statistic alarm thresholds.

When thresholds are exceeded, the system generates alarms that indicate the possible deterioration of an E1 link. Refer to the following parameters to determine the specific E1 data type that needs to be configured. You can define one alarm threshold for each parameter.

parameter	definition
number	Statistic alarm threshold number (1 - 10).
variable	Variable on which a threshold is to be configured
ses	Threshold for Severely Errored Seconds
es	Threshold for Errored Seconds
bes	Threshold for Bursty Errored Seconds
uas	Threshold for Unavailable Seconds
eev	Threshold for Excessive Error Violation Seconds
lofc	Threshold for Loss-of-Frame Counts
css	Threshold for Controlled Slip Seconds
oof	Threshold for Out-of-Frame Seconds
crc	Threshold for CRC-6 errors
bpv	Threshold for Bipolar Violations
interval	Sampling interval, in seconds. The range is 1 - 65535.
rising_threshold	Number of errored seconds or events which, if exceeded during any sampling interval, results in a rising alarm. The range is 0 - 2147483647.
falling_threshold	Minimum number of errored seconds or events below which a falling alarm is reported. This alarm is reported if a rising alarm was previously reported and the number of errored seconds or events subsequently dropped below this minimum threshold. The falling threshold value must be less than the rising threshold value above. The range is 0 - 2147483647.
sampling_type	Method of sampling, as follows:
absolute	The errored second or event count is compared directly to the specified threshold values, and the appropriate alarm type (rising or falling) is reported.
delta	The errored second or event count is compared to the difference between the rising and falling thresholds above, and a rising alarm is reported if the actual error count exceeds that difference. This is the default setting if you do not specify a sampling type.

syntax:

user < 1 - 10 > < ses |es | bes | uas | eev | lofc |css |oof | crc | bpv > < 1 - 65535> < 0 - 2147483647 > < 0 - 2147483647 > [< absolute | delta >]

example:

RouterE/configure/module/e1 1-3/alarms/thresholds> **user 1 ses 300 50 30**

In this example, the monitored parameter is the number of Severely Errored Seconds that occur. The sampling interval is set to 300 seconds (5 minutes). The rising alarm threshold is 50 SES and the falling alarm threshold is 30 SES. Delta sampling is used (by default), in which a rising alarm is reported if the SES count increases by 20 (50 - 30 = 20) during any sampling interval. Similarly, a falling alarm is reported if the SES count decreases by 20 during any interval.

applicable models:

OmniAccess 601, OmniAccess 602E

configure module e1 carrier-type

This command configures the type of carrier being used. Select either T1 or E1.

parameter	definition
t1	Specifies a T1 carrier.
e1	Specifies an E1 carrier.

syntax:

carrier-type < t1 | e1 >

example:

Router/configure/module/e1> **carrier_type t1**

related commands:

configure system licenses
configure system logging
configure system mac_range
configure system routing

applicable models:

OmniAccess 601

configure module e1 circuit_Id

This command specifies an optional circuit name to the E1 channel.

parameter	definition
ckt_id	Optional circuit name for the E1 channel.

syntax:

circuit_Id ckt_id < *name* >

example:

RouterE/configure/module/e1 1-3> **circuit_Id Main01**

related commands:

configure module e1 alarms
configure module e1 clock_source
configure module e1 contactInfo
configure module e1 description
configure module e1 enable
configure module e1 framing
configure module e1 jitter

applicable models:

OmniAccess 601, OmniAccess 602E

configure module e1 clock_source

This command specifies a network timing source for an E1 link.

parameter	definition
clock_source	The desired clock source
internal	Router system's internal clock (default).
line	Clock recovered from the incoming E1 signal (loop timing).

syntax:

clock_source clock_source < internal | line >

example:

RouterE/configure/module/e1 1> **clock_source line**

related commands:

configure module e1 alarms
configure module e1 circuit_Id
configure module e1 contactInfo
configure module e1 description
configure module e1 enable
configure module e1 framing
configure module e1 jitter

applicable models:

OmniAccess 601, OmniAccess 602E

configure module e1 contactInfo

This command specifies a person to contact for information regarding the E1 link.

parameter	definition
contactInfo	Person to contact for information.

syntax:

contactInfo contactinfo < *name* >

example:

RouterE/configure/module/e1 1-3> **contactInfo James Smythe**

related commands:

configure module e1 alarms
configure module e1 circuit_Id
configure module e1 clock_source
configure module e1 description
configure module e1 enable
configure module e1 framing
configure module e1 jitter

applicable models:

OmniAccess 601, OmniAccess 602E

configure module e1 description

This command describes the E1 interface.

parameter	definition
description	Describes or names the E1 interface

syntax:

description description < *name* >

example:

RouterE/configure/module/e1-1> **xyzInc**

related commands:

configure module e1 alarms
configure module e1 circuit_Id
configure module e1 clock_source
configure module e1 contactInfo
configure module e1 enable
configure module e1 framing
configure module e1 jitter

applicable models:

OmniAccess 601, OmniAccess 602E

configure module e1 enable

This command places an E1 link in service and allows the link to transmit and receive data.

To take an E1 link out of service, type **no enable** at the configure/module/e1> prompt. This action sends an all-ones Alarm Indication Signal (AIS) to the far end and places the link out of service.

To verify the in-service or out-of-service status of an E1 channel, use the **display module configuration e1** command.

syntax:

[no] enable

example:

RouterE/configure/module/e1 1-3> **enable**

related commands:

configure module e1 alarms
configure module e1 circuit_Id
configure module e1 clock_source
configure module e1 contactInfo
configure module e1 description
configure module e1 framing
configure module e1 jitter

applicable models:

OmniAccess 601, OmniAccess 602E

configure module e1 framing

This command sets the framing mode for an E1 link.

parameter	definition
framing	
crc (default)	Sets the framing to CRC. Standard CRC-4 error detection as defined by ITUT.
noncrc	Sets the framing to non-CRC.
disable	Disables framing.

syntax:

framing framing < crc | noncrc | disable >

example:

RouterE/configure/module/e1 1-3> **framing noncrc**

related commands:

configure module e1 alarms
configure module e1 circuit_Id
configure module e1 clock_source
configure module e1 contactInfo
configure module e1 description
configure module e1 enable
configure module e1 jitter

applicable models:

OmniAccess 601, OmniAccess 602E

configure module e1 jitter

This command configures the E1 interface to eliminate any frequency variation within the data stream.

parameter	definition
enable	Enables the jitter attenuation capability (default).
disable	Disables the jitter attenuation capability.

syntax:

jitter < enable | disable >

example:

RouterE/configure/module/e1 1-3> **jitter disable**

related commands:

configure module e1 alarms
configure module e1 circuit_Id
configure module e1 clock_source
configure module e1 contactInfo
configure module e1 description
configure module e1 enable
configure module e1 framing

applicable models:

OmniAccess 601, OmniAccess 602E

configure module t1

This command accesses next-level commands for selecting a T1 link for configuration.

To select multiple contiguous T1 links for configuration, type **t1 n-n** (where **n-n** is the range of T1s to be configured).

parameter	definition
t1_no	Number of the T1 WAN link to be configured. The range is 1 - 16, depending on the system model.

syntax:

module t1 t1_no < n >

example:

Router/configure> **module t1 1**

next-level commands

configure module t1 alarms
configure module t1 circuitID
configure module t1 clock_source
configure module t1 contactInfo
configure module t1 description
configure module t1 enable
configure module t1 fdl
configure module t1 framing
configure module t1 linecode
configure module t1 linemode
configure module t1 yellow_alarm

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 alarms

This command accesses next-level commands for configuring alarm reporting on a T1 link.

syntax:

alarms

example:

Router/configure/module/t1 1> **alarms**

next-level commands

configure module t1 alarms thresholds
configure module t1 alarms hierarchy

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 alarms hierarchy

This command enables and disables the hierarchy for displaying RLOS and RLOF on a T1 link. Users may select an individual T1 or a range of T1s when using this command.

syntax:

alarms hierarchy

example:

Router/configure/module/t1 1-3> **alarms hierarchy**

To view the alarms hierarchy once it has been set, use the **display module configuration** command. The display will indicate either "True" or "False." True means that RLOS is RLOS only; False means that RLOS is RLOS plus RLOF.

related commands:

configure module t1 alarms thresholds

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 alarms thresholds

This command accesses next-level commands for configuring alarm thresholds.

If a user-defined threshold is exceeded, the system reports an alarm. These alarms indicate the possible deterioration of an associated T1link.

syntax:

thresholds

example:

Router/configure/module/t1 1-3/alarms> **thresholds**

next-level commands

configure module t1 alarms thresholds user

related commands:

configure module t1 alarms hierarchy

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 alarms thresholds user

This command sets the T1 user statistic alarm thresholds.

When thresholds are exceeded, the system generates alarms that indicate the possible deterioration of a T1 link. Refer to the following parameters to determine the specific T1 data type that needs to be configured. You can define one alarm threshold for each parameter.

parameter	definition
number	Statistic alarm threshold number The range is 1 - 10.
variable	Variable on which a threshold is to be configured.
ses	Threshold for Severely Errored Seconds
es	Threshold for Errored Seconds
bes	Threshold for Bursty Errored Seconds
uas	Threshold for Unavailable Seconds
eev	Threshold for Excessive Error Violation Seconds
lofc	Threshold for Loss-of-Frame Counts
css	Threshold for Controlled Slip Seconds
oof	Threshold for Out-of-Frame Seconds
crc	Threshold for CRC-6 errors
bpv	Threshold for Bipolar Violations
interval	Sampling interval, in seconds. The range is 1 - 65535.
rising_threshold	Number of errored seconds or events which, if exceeded during any sampling interval, results in a rising alarm. The range is 0 - 2147483647.
falling_threshold	Minimum number of errored seconds or events below which a falling alarm is reported. This alarm is reported if a rising alarm was previously reported and the number of errored seconds or events subsequently dropped below this minimum threshold. The falling threshold value must be less than the rising threshold value above. The range is 0 - 2147483647.
sampe_type	Method of sampling, as follows:
absolute	The errored second or event count is compared directly to the specified threshold values, and the appropriate alarm type (rising or falling) is reported.
delta	The errored second or event count is compared to the difference between the rising and falling thresholds above, and a rising alarm is reported if the actual error count exceeds that difference. This is the default setting if you do not specify a sampling type.

syntax:

user < 1 - 10 > < ses | es | bes | uas | eev | lofc | css | oof | crc | bpv >
< 1 - 65535 > < 0 - 2147483647 > < 0 - 2147483647 > [< absolute | delta >]

example:

Router/configure/module/t1 1-3/alarms/thresholds> **user 1 ses 300 50 30**

In this example, the monitored parameter is the number of Severely Errored Seconds that occur. The sampling interval is set to 300 seconds (5 minutes). The rising alarm threshold is 50 SES and the falling alarm threshold is 30 SES. Delta sampling is used (by default), in which a rising alarm is reported if the SES count increases by 20 (50 - 30 = 20) during any sampling interval. Similarly, a falling alarm is reported if the SES count decreases by 20 during any interval.

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 circuit_Id

This command specifies an optional circuit name for a T1 channel.

parameter	definition
ckt_id	Optional circuit name for the T1 channel

syntax:

circuit_Id ckt_id < *name* >

example:

Router/configure/module t1 1-3> **circuit_Id Main01**

related commands:

configure module t1 alarms
configure module t1 clock_source
configure module t1 contactInfo
configure module t1 description
configure module t1 enable
configure module t1 fdl
configure module t1 framing
configure module t1 linecode
configure module t1 linemode
configure module t1 yellow_alarm

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 clock_source

This command specifies a network timing source for a T1 link.

parameter	definition
clock_source	The desired clock source
internal	Router system's internal clock (default)
line	Clock recovered from the incoming T1 signal (loop timing)

syntax:

clock_source clock_source < internal | line >

example:

Router/configure/module/t1 1-3> **clock_source line**

related commands:

configure module t1 alarms
configure module t1 circuitID
configure module t1 contactInfo
configure module t1 description
configure module t1 enable
configure module t1 fdl
configure module t1 framing
configure module t1 linecode
configure module t1 linemode
configure module t1 yellow_alarm

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 contactInfo

This command specifies a person to contact for information regarding the T1 link.

parameter	definition
contactInfo	Person to contact for information

syntax:

contactInfo contactinfo < *name* >

example:

Router/configure/module t1 1-3> **contactInfo James Smythe**

related commands:

configure module t1 alarms
configure module t1 circuitID
configure module t1 clock_source
configure module t1 description
configure module t1 enable
configure module t1 fdl
configure module t1 framing
configure module t1 linecode
configure module t1 linemode
configure module t1 yellow_alarm

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 description

This command describes the T1 interface.

parameter	definition
description	Describes or names the T1 interface

syntax:

description description < *name* >

example:

Router/configure/module/t1-1> **xyzInc**

related commands:

configure module t1 alarms
configure module t1 circuitID
configure module t1 clock_source
configure module t1 contactInfo
configure module t1 enable
configure module t1 fdl
configure module t1 framing
configure module t1 linecode
configure module t1 linemode
configure module t1 yellow_alarm

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 enable

This command places a T1 link in service and allows the link to transmit and receive data.

To take a T1 link out of service, type **no enable** at the configure/module/t1> prompt. This action sends an all-ones Alarm Indication Signal (AIS) to the far end and places the link out of service.

To verify the in-service or out-of-service status of a T1 channel, use the **display module configuration t1** command.

syntax:

[no] enable

example:

Router/configure/module/t1 1-3> **enable**

related commands:

configure module t1 alarms
configure module t1 circuitID
configure module t1 clock_source
configure module t1 contactInfo
configure module t1 description
configure module t1 fdl
configure module t1 framing
configure module t1 linecode
configure module t1 linemode
configure module t1 yellow_alarm

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 fdl

This command selects the FDL protocol for an ESF-framed T1 link.

Before using this command, set the desired T1 link framing mode to ESF using the **configure module t1 framing** command.

parameter	definition
fdl_type	
ansi_only	ANSI T1.403 support
att_only	AT&T TR54016 support
ansi_att	Both ANSI & ATT support (default)
c_d_su	
csu	CSU type
dsu	DSU type
csu_dsu	CSUDSU type (default)

syntax:

fdl fdl_type < ansi | att | ansi_att >[c_d_su < csu | dsu | csu_dsu >]

example:

Router/configure/module/t1 1-3> **fdl ansi_att**

In this example, the FDL is configured for both ANSI and AT&T data.

related commands:

configure module t1 alarms
configure module t1 circuitID
configure module t1 clock_source
configure module t1 contactInfo
configure module t1 description
configure module t1 enable
configure module t1 framing
configure module t1 linecode
configure module t1 linemode
configure module t1 yellow_alarm

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 framing

This command sets the framing mode for a T1 link.

parameter	definition
framing	
esf	Extended Super Frame framing format for T1 (default)
d4	Super Frame framing format for T1

syntax:

framing framing < esf | d4 >

example:

Router/configure/module/t1 1-3> **framing d4**

related commands:

configure module t1 alarms
configure module t1 circuitID
configure module t1 clock_source
configure module t1 contactInfo
configure module t1 description
configure module t1 enable
configure module t1 fdl
configure module t1 linecode
configure module t1 linemode
configure module t1 yellow_alarm

applicable models:

All models.

NOTE: Do not use the all zeros pattern when using AMI mode on D4 and ESF framing.

configure module t1 linecode

This command sets the type of line coding for a T1 link.

parameter	definition
linecode	
b8zs	B8ZS linecode for T1 (default)
amI	AMI linecode for T1

syntax:

linecode linecode < b8zs | ami >

example:

Router/configure/module/t1 1-3> **linecode b8zs**

related commands:

configure module t1 alarms
configure module t1 circuitID
configure module t1 clock_source
configure module t1 contactInfo
configure module t1 description
configure module t1 enable
configure module t1 fdl
configure module t1 framing
configure module t1 linemode
configure module t1 yellow_alarm

applicable models:

All models.

configure module t1 linemode

This command accesses next-level commands for defining the type of T1 WAN interface and setting the T1 signal level accordingly.

syntax:

linemode

example:

Router/configure/module/t1 1-3> **linemode**

next-level commands

configure module t1 linemode csu
configure module t1 linemode dsx

related commands:

configure module t1 alarms
configure module t1 circuitID
configure module t1 clock_source
configure module t1 contactInfo
configure module t1 description
configure module t1 enable
configure module t1 fdl
configure module t1 framing
configure module t1 linecode
configure module t1 yellow_alarm

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 linemode csu

This command sets the amount of T1 line build-out for a CSU interface.

parameter	definition
lbo	
db_zero	Configure LBO for zero db (default)
db7_5	Configure LBO for 7.5 db
db15	Configure LBO for 15 db
db22_5	Configure LBO for 22.5 db

syntax:

csu lbo < db_zero | db7_5 | db15 | db22_5 >

example:

Router/configure/module/t1 1-3/linemode> **csu db15**

related commands:

configure module t1 linemode dsx

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 linemode dsx

This command sets the amount of T1 signal equalization based on the cabling distance to the DSX cross-connect.

parameter	definition
cable_length	Cabling distance to DSX bay, as follows:
1	0 - 133 feet
2	133 - 266 feet
3	266 - 399 feet
4	399 - 533 feet
5	533 - 655 feet

syntax:

dsx cable_length < 1 | 2 | 3 | 4 | 5 >

example:

Router/configure/module/t1 1-3/linemode> **dsx 2**

related commands:

configure module t1 linemode csu

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure module t1 yellow_alarm

This command sets the yellow alarm operation on a T1 link.

parameter	definition
yellow_alarm	Type of yellow alarm operation
generate	Generate and send yellow alarms to network.
detect	Detect incoming yellow alarms from network.
gen_det	Generate and detect yellow alarms.
disable	Disable yellow alarm generator and detector.

syntax:

yellow_alarm yellow_alarm < generate | detect | gen_det | disable >

example:

Router/configure/module/t1 1-3> **yellow_alarm gen_det**

related commands:

configure module t1 alarms
configure module t1 circuitID
configure module t1 clock_source
configure module t1 contactInfo
configure module t1 description
configure module t1 enable
configure module t1 fdl
configure module t1 framing
configure module t1 linecode
configure module t1 linemode

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

configure network

This command sets the Router system parameters from a configuration file on a network server.

This method downloads files via the TFTP protocol, then executes the commands from that file without operator intervention. Use the **configure network** command if you want to configure one or more Router systems with the same configuration.

syntax:

configure network

example:

Router> **configure network**

After you enter the **configure network** command, press **Return**, follow the prompt (as shown below), and enter the configuration information.

host: **NewYork**

configuration file (path): **/networks/system01.cfg**

related commands:

configure flash
configure terminal

applicable models:

All models.

configure qos

This command accesses next-level commands for configuring QoS.

syntax:

qos

example:

Router/configure> **qos**

next-level commands

configure qos add_class class_name parent
configure qos historical_stats
configure qos REM
configure qos REM_

applicable models:

All models.

configure qos add_class class_name parent

This command loads QoS class templates containing CR and BR settings from a specified file.

A maximum of 36 templates can be defined in the system.

parameter	definition
class_name	The name assigned to a new QoS class. The maximum number of characters allowed is 19.
parent	Name of the parent class; QoS classes are defined hierarchically. To add to the root, specify either **root-in** or **root-out.**
cr	Committed Rate (entered in Kbps). Minimum value = 1 Kbps. Maximum value = interface bandwidth. The sum of CRs of all classes at a specific level cannot exceed BR. Supported for outbound traffic only.
br	Burst Rate (entered in Kbps). Minimum value = 1 Kbps. Maximum value = interface bandwidth or the BR value of the parent class, whichever is lesser. BR >= CR for a class. Supported for outbound traffic only.
priority	Scheduling priority. Valid only for leaf classes. The range is 1 - 8. Supported for outbound traffic only.
src_ip_address	Source IP address, range of source IP addresses, source subnet, or (default) to which QoS parameters will apply.
dst_ip_address	Destination IP address, range of destination IP addresses, destination subnet, or (default) to which QoS parameters will apply.
netmask	Subnet mask used to specify a source or destination subnet to which QoS parameters will be applied.
port	Port number or (default) indicating which applications QoS parameters will be applied to.
vlan_id	The VLAN ID number or range of VLAN ids to which the QoS parameters will apply. The range is 1 - 4095 or default.
dscp	DiffServ codepoint or a range of DiffServ codepoints to which the QoS parameters will apply. The range is 0 - 63.
dot1p <priority value>	Specifies 802.1p protocol. Priority value range is 0 to 7.
nat_ip	Specifies a NAT translation address for this class.
mark_dscp	Marks all packets with a specific DiffServ codepoint. The range is 0 - 63.
mark_vlan	Tags packets with specific VLAN id if tagging is enabled
mark_dot1p	Tags packets with specific 802.1p priority value. Valid priority value range is 0-7.

syntax:

add_class class_name parent [cr] [br] [priority] [src_ip_address] [dst_ip_address] [netmask] [port] [vlan_id] [dscp] [dot1p] [nat_ip] [mark_dscp] [mark_vlan] [mark_dot1p]<cr>

example:

Router/configure/qos> **configure qos add_class class_name parent class_name test**

related commands:

configure qos bulk_statistics

configure qos bulk_stats_ftp

configure qos add_class class_name parent

configure qos REM

configure qos REM_

applicable models:

All models.

configure qos historical_stats

This command accesses next-level commands for configuring historical statistics variables.

syntax:

historical_stats

example:

Router/configure/qos> **historical_stats**

next-level commands

configure qos historical_stats ftp_parameters
configure qos historical_stats sample_interval
configure qos historical_stats upload

applicable models:

All models.

configure qos historical_stats ftp_parameters

This command configures ftp parameters related to historical statistics upload.

It is an interactive command and prompts for the primary server's IP address, secondary server's IP address (optional), the user name, and password ID. The configuration is stored in NVRAM and not in a configuration file, therefore, the specified values are stored across reboots.

syntax:

ftp_parameters

example:

Router/configure/qos/historical_stats> **ftp_parameters**

The following screen display example shows the prompt sequence that occurs upon execution of this command.

screen display example

```
T6300/configure/qos/historical_stats> ftp_parameters
Primary Ftp Server: 10.1.4.9
Secondary Ftp Server: 10.1.4.10
Ftp user name: jdoe
Ftp password:
```

related commands:

configure qos historical_stats sample_interval
configure qos historical_stats upload

applicable models:

All models.

configure qos historical_stats sample_interval

This command configures the length of the historical statistics sample interval.

Changing the sample interval will restart the historical stats collection process, and all accumulated historical stats will be lost.

parameter	definition
interval	The length of the sample interval in minutes. Enter a number: 5, 10, or 15).

syntax:

sample_interval interval < n >

example:

Router/configure/qos/historical_stats> **sample interval 15**

related commands:

configure qos historical_stats ftp_parameters
configure qos historical_stats upload

applicable models:

All models.

configure qos historical_stats upload

This command enables and configures uploading of historical statistics.

This command enables uploading of historical stats and configures the upload interval and fileid. The fileid is an identifier string that is used to form the upload file name along with the upload date and time. If this string is not configured, the IP address of Ethernet 0 is used instead to identify the Router system uploading the file.

parameter	definition
interval	Upload interval in hours
	Enter a number: 1, 2, 3, or 4; The value "1" is the default.
fileID	Upload file ID
	Enter a word.

syntax:

upload [interval < n >] [fileID]

example:

Router/configure/qos/historical_stats> **upload 2 taz123**

related commands:

configure qos historical_stats ftp_parameters
configure qos historical_stats sample_interval

applicable models:

All models.

configure qos REM

This command allows users to add comments (at the beginning of the QoS area of the configuration file) during a configuration session.

Comments will appear after using the **save local** command.

parameter	definition
comments	80-character (maximum) string enclosed in quotation marks.

syntax:

REM comments < "comments" >

example:

Router/configure/qos> **REM "QoS configured on November 4, 2000.TVD 11:30."**

related commands:

configure qos add_class class_name parent
configure qos REM_
display configuration stored

applicable models:

All models.

configure qos REM_

This command allows users to add comments (at the end of the QoS area of the configuration file) during a configuration session.

Comments will appear after using the **write memory** command.

parameter	definition
comments	80-character (maximum) string enclosed in quotation marks.

syntax:

REM_ comments < "comments" >

example:

host/configure/qos> **REM_ "QoS policy reviewed on November 4, 2000.TVD 11:45."**

related commands:

configure qos bulk_statistics
configure qos bulk_stats_ftp
configure qos add_class class_name parent
configure qos REM
display configuration stored

applicable models:

All models.

configure reverse_telnet

This command accesses next-level commands for configuring reverse telnet.

syntax:

reverse_telnet

example:

Router-OmniAccess 604/configure> **reverse_telnet**

next-level commands

configure reverse_telnet enable
configure reverse_telnet set_baud_rate
configure reverse_telnet set_data_bits
configure reverse_telnet set_flow_control
configure reverse_telnet set_parity
configure reverse_telnet set_stop_bits
configure reverse_telnet telnet_port
configure reverse_telnet telnet_timeout

applicable models:

All models.

configure reverse_telnet enable

This command enables reverse telnet on the router.

syntax:

enable

example:

Router-OmniAccess 604/configure/reverse_telnet> **enable**

related commands:

configure reverse_telnet set_baud_rate
configure reverse_telnet set_data_bits
configure reverse_telnet set_flow_control
configure reverse_telnet set_parity
configure reverse_telnet set_stop_bits
configure reverse_telnet telnet_port
configure reverse_telnet telnet_timeout

applicable models:

All models.

configure reverse_telnet set_baud_rate

This command sets the baud rate for the second serial port.

parameter	definition
baud_rate	The range is 50 - 115200; the default is 9600.

syntax:

set_baud_rate baud_rate < n >

example:

Router-OmniAccess 604/configure/reverse_telnet> **set_baud_rate 65**

related commands:

configure reverse_telnet enable
configure reverse_telnet set_data_bits
configure reverse_telnet set_flow_control
configure reverse_telnet set_parity
configure reverse_telnet set_stop_bits
configure reverse_telnet telnet_port
configure reverse_telnet telnet_timeout

applicable models:

All models.

configure reverse_telnet set_data_bits

This command sets the data bits for the second serial port.

parameter	definition
data_bits	The number of data bits configured. The range is 5 - 8; the default is 8.

syntax:

set_data_bits data_bits < n >

example:

Router-OmniAccess 604/configure/reverse_telnet> **set_data_bits 5**

related commands:

configure reverse_telnet enable
configure reverse_telnet set_baud_rate
configure reverse_telnet set_flow_control
configure reverse_telnet set_parity
configure reverse_telnet set_stop_bits
configure reverse_telnet telnet_port
configure reverse_telnet telnet_timeout

applicable models:

All models.

configure reverse_telnet set_flow_control

This command sets the flow control for the second serial port.

parameter	definition
flow_control	
0	No flow control. The default is 0.
1	Software flow control.
2	Hardware flow control.

syntax:

set_flow_control flow_control < n >

example:

Router-OmniAccess 604/configure/reverse_telnet> **set_flow_control 1**

related commands:

configure reverse_telnet enable
configure reverse_telnet set_baud_rate
configure reverse_telnet set_data_bits
configure reverse_telnet set_parity
configure reverse_telnet set_stop_bits
configure reverse_telnet telnet_port
configure reverse_telnet telnet_timeout

applicable models:

All models.

configure reverse_telnet set_parity

This command sets the parity for the second serial port.

parameter	definition
parity	
0	No parity is configured. The default is 0.
1	Odd parity is configured.
2	Even parity is configured.

syntax:

set_parity parity < n >

example:

Router-OmniAccess 604/configure/reverse_telnet> **set_parity 0**

related commands:

configure reverse_telnet enable
configure reverse_telnet set_baud_rate
configure reverse_telnet set_data_bits
configure reverse_telnet set_flow_control
configure reverse_telnet set_stop_bits
configure reverse_telnet telnet_port
configure reverse_telnet telnet_timeout

applicable models:

All models.

configure reverse_telnet set_stop_bits

This command configures the stop bits for the second serial port.

parameter	definition
stop_bits	The number of stop bits to configure. The range is 1 - 2; the default is1.

syntax:

set_stop_bits stop_bits < n >

example:

Router-OmniAccess 604/configure/reverse_telnet> **set_stop_bits 2**

related commands:

configure reverse_telnet enable
configure reverse_telnet set_baud_rate
configure reverse_telnet set_data_bits
configure reverse_telnet set_flow_control
configure reverse_telnet set_parity
configure reverse_telnet telnet_port
configure reverse_telnet telnet_timeout

applicable models:

All models.

configure reverse_telnet telnet_port

This command configures the telnet port.

parameter	definition
port	The number of the port to be configured for telnet. The range is 2000 - 65535; the default is 2001.

syntax:

telnet_port < n >

example:

Router-OmniAccess 604/configure/reverse_telnet> **telnet_port 3500**

related commands:

configure reverse_telnet enable
configure reverse_telnet set_baud_rate
configure reverse_telnet set_data_bits
configure reverse_telnet set_flow_control
configure reverse_telnet set_parity
configure reverse_telnet set_stop_bits
configure reverse_telnet telnet_timeout

applicable models:

All models.

configure reverse_telnet telnet_timeout

This command configures the timeout (in seconds) for the telnet connection.

parameter	definition
timeout	The timeout period in seconds. The range is 0 -7200; the default is 600. Enter 0 to disable telnet timeout.

syntax:

telnet_timeout timeout < n >

example:

Router-OmniAccess 604/configure/reverse_telnet> **telnet_timeout 0**

related commands:

configure reverse_telnet enable
configure reverse_telnet set_baud_rate
configure reverse_telnet set_data_bits
configure reverse_telnet set_flow_control
configure reverse_telnet set_parity
configure reverse_telnet set_stop_bits
configure reverse_telnet telnet_port

applicable models:

All models.

configure router

Enables the specified routing protocol. Use this command to enter the specified router configuration mode. Refer to the *User Guide* for examples.

parameter	definition
bgp	Enables Border Gateway Protocol.
ospf	Enables Open Shortest Path First protocol.
rip	Enables Routing Information Protocol.
routerid	Specifies the IP address of the router.

syntax:

COMMANDS

example:

host/configure> **router rip**

applicable models:

All models.

configure secure_passwords

This command specifies that passwords stored in the configuration file be encrypted.

syntax:

secure_passwords

example:

Router/configure> secure_passwords

applicable models:

All models.

configure snmp-server

This command accesses next-level commands for configuring the Router MIB database (industry and enterprise) for all systems.

syntax:

snmp-server

example:

Router/configure> **snmp-server**

next-level commands

configure snmp-server community
configure snmp-server contact
configure snmp-server enable traps
configure snmp-server location
configure snmp-server snmp-source
configure snmp-server REM
configure snmp-server REM_
configure snmp-server trap-source
configure snmp-server chassis-id
configure snmp-server trap-host

applicable models:

All models.

configure snmp-server community

This command sets the SNMP community name and access privileges.
This entry is a password string that assigns access privileges to the Router system SNMP agent.

parameter	definition
community	SNMP MIB community name Use up to 64 characters.
access_privilege	
ro	Read-only (can get but not set MIB parameters)
rw	Read/write (can get and set MIB parameters)

syntax:
community < *name* > access_privilege < ro | rw >

example:

Router/configure/snmp> **community network01 rw**

related commands:

configure snmp-server contact
configure snmp-server enable traps
configure snmp-server location
configure snmp-server REM
configure snmp-server REM_
configure snmp-server trap-source
configure snmp-server chassis-id
configure snmp-server trap-host

applicable models:

All models.

configure snmp-server contact

This command specifies a contact person for the SNMP MIB.

parameter	definition
contact	Name of the person to contact regarding the SNMP MIB, enclosed in quotation marks. Use up to 244 characters and enclose in quotes.

syntax:

contact contact < *"name"* >

example:

Router/configure/snmp> **contact "JoeSmith"**

related commands:

configure snmp-server community
configure snmp-server enable traps
configure snmp-server location
configure snmp-server REM
configure snmp-server REM_
configure snmp-server trap-source
configure snmp-server chassis-id
configure snmp-server trap-host

applicable models:

All models.

configure snmp-server location

This command defines the SNMP host system location.

parameter	definition
string	Location of the host system

syntax:

location location < *string* >

example:

Router/configure/snmp> **location fremont_ca**

related commands:

configure snmp-server community
configure snmp-server contact
configure snmp-server enable traps
configure snmp-server REM
configure snmp-server REM_
configure snmp-server trap-source
configure snmp-server chassis-id
configure snmp-server trap-host

applicable models:

All models.

configure snmp-server REM

This command allows users to add comments (at the beginning of the SNMP area of the configuration file) during a configuration session.

Comments will appear in the configuration file after using the **save local** command.

parameter	definition
comments	80-character (maximum) string enclosed in quotation marks.

syntax:

REM < "string" >

example:

Router/configure/snmp> **REM "SNMP configured on November 4, 2000.TVD 11:00"**

related commands:

configure snmp-server community
configure snmp-server contact
configure snmp-server enable traps
configure snmp-server location
configure snmp-server REM_
configure snmp-server trap-source
configure snmp-server chassis-id
configure snmp-server trap-host
display configuration stored

applicable models:

All models.

configure snmp-server REM_

This command allows users to add comments (at the end of the SNMP area of the configuration file) during a configuration session.

Comments will appear in the configuration file after using the **save local** command.

parameter	definition
comments	80-character (maximum) string enclosed in quotation marks.

syntax:

REM_ comments < "string" >

example:

Router/configure/snmp> **REM_ "SNMP configured on November 4, 2000.TVD 11:00."**

related commands:

configure snmp-server community
configure snmp-server contact
configure snmp-server enable traps
configure snmp-server location
configure snmp-server REM
configure snmp-server trap-source
configure snmp-server chassis-id
configure snmp-server trap-host
display configuration stored

applicable models:

All models.

configure snmp-server chassis-id

This command names the SNMP host system.

parameter	definition
id	Name of the host system

syntax:

chassis-id id < *name* >

example:

Router/configure/snmp-server> **chassis-id sanjose_ca**

related commands:

configure snmp-server community
configure snmp-server contact
configure snmp-server enable traps
configure snmp-server location
configure snmp-server REM
configure snmp-server REM_
configure snmp-server trap-source
configure snmp-server trap-host

applicable models:

All models.

configure snmp-server enable traps

This command accesses next-level commands for configuring SNMP traps.

Once these traps are enabled, the system sends all configured traps to all SNMP trap hosts previously configured via the **configure snmp-server trap-host** command. To view the currently configured SNMP traps, use the **display snmp traps** command.

NOTE: Bundle events, such as *bundle up* or *bundle down*, do not produce SNMP traps. Physical layer events do generate traps and should be monitored for status of WAN bundles.

syntax:

enable traps

example:

Router/configure/snmp-server> **enable_trap**

next-level commands

configure snmp-server enable traps bgp
configure snmp-server enable traps config
configure snmp-server enable traps environment
configure snmp-server enable traps failover
configure snmp-server enable traps frame_relay
configure snmp-server enable traps ospf
configure snmp-server enable traps snmp
configure snmp-server enable traps sntp
configure snmp-server enable traps system
configure snmp-server enable traps vrrp

applicable models:

All models.

configure snmp-server enable traps bgp

This command enables or disables BGP-related traps.

parameter	definition
trap_est	
established	Enables or disables BGP established notification trap
trap_back	
backward	Enables of disables BGP backward transition notification trap

syntax:

bgp [trap_est < established >] [trap_back < backward >]

example:

Router/configure/snmp/enable_trap> **bgp**

related commands:

configure snmp-server enable traps config
configure snmp-server enable traps environment
configure snmp-server enable traps failover
configure snmp-server enable traps frame_relay
configure snmp-server enable traps ospf
configure snmp-server enable traps snmp
configure snmp-server enable traps sntp
configure snmp-server enable traps system
configure snmp-server enable traps vrrp

applicable models:

All models.

operating mode:

Routing only.

configure snmp-server enable traps config

This command enables or disables traps for configuration changes and saves.

parameter	definition
trap_change	
change	Enables a system configuration trap.
trap_save	
save	Enables a save trap.

syntax:

[no] config [trap_change < change >] [trap_save < save >]

example:

Router/configure/snmp/enable_trap> **config change**

related commands:

configure snmp-server enable traps bgp
configure snmp-server enable traps environment
configure snmp-server enable traps failover
configure snmp-server enable traps frame_relay
configure snmp-server enable traps ospf
configure snmp-server enable traps snmp
configure snmp-server enable traps sntp
configure snmp-server enable traps system
configure snmp-server enable traps vrrp

applicable models:

All models.

configure snmp-server enable traps environment

This command enables and disables traps for fan failure and/or temperature level changes.

parameter	definition
trap_fan	
fan	Enables fan failure trap
trap_temp	
temperature	Enables temperature level change trap
trap_power	
power	Enables power supply status change trap

syntax:

[no] environment [trap_fan < fan >] [trap_temp < temperature >]
[trap_power < power >]

example:

Router/configure/snmp/enable_trap> **environment fan**

related commands:

configure snmp-server enable traps bgp
configure snmp-server enable traps config
configure snmp-server enable traps failover
configure snmp-server enable traps frame_relay
configure snmp-server enable traps ospf
configure snmp-server enable traps snmp
configure snmp-server enable traps sntp
configure snmp-server enable traps system
configure snmp-server enable traps vrrp

applicable models:

All models.

configure snmp-server enable traps failover

This command enables or disables traps for both successful and unsuccessful failovers.

parameter	definition
trap_success success	Enables or disables a fail over successful notification trap
trap_failure failure	Enables or disables a fail over failed notification trap

syntax:

[no] failover [trap_success < success >] [trap_failure < failure >]

example:

Router/configure/snmp/enable_trap> **no failover failure**

related commands:

configure snmp-server enable traps bgp
configure snmp-server enable traps config
configure snmp-server enable traps environment
configure snmp-server enable traps frame_relay
configure snmp-server enable traps ospf
configure snmp-server enable traps snmp
configure snmp-server enable traps sntp
configure snmp-server enable traps system
configure snmp-server enable traps vrrp

applicable models:

All models.

configure snmp-server enable traps frame_relay

This command enables or disables traps for frame relay virtual circuit state changes.

parameter	definition
trap_vcstate	
vcstate	Enables or disables VC state change trap

syntax:

[no] frame_relay [trap_vcstate < vcstate >]

example:

Router/configure/snmp/enable_trap> **frame_relay vcstate**

related commands:

configure snmp-server enable traps bgp
configure snmp-server enable traps config
configure snmp-server enable traps environment
configure snmp-server enable traps failover
configure snmp-server enable traps ospf
configure snmp-server enable traps snmp
configure snmp-server enable traps sntp
configure snmp-server enable traps system
configure snmp-server enable traps vrrp

applicable models:

All models.

configure snmp-server enable traps ospf

This command enables or disables OSPF related traps.

parameter	definition
if_state ifStateChange	Enables or disables OSPF interface state change notification trap
virt_state virtIfStateChange	Enables or disables OSPF virtual interface state change notification trap
nbr_state nbrStateChange	Enables or disables OSPF neighbor state change notification trap
vnbr_state VirtnbrStateChange	Enables or disables OSPF virtual neighbor state change notification trap
if_error IfConfigError	Enables or disables OSPF interface configuration error notification trap
virt_error virtIfConfigError	Enables or disables OSPF virtual interface configuration error notification trap
if_auth ifAuthFailure	Enables or disables OSPF interface authentication failure notification trap
virt_auth virtIfAuthFailure	Enables or disables OSPF virtual inteface authentication failure notification trap
if_rxbad ifRxBadpacket	Enables or disables OSPF interface receive bad packet notification trap
virt_rxbad virtIfRxBadpacket	Enables or disables OSPF virtual interface receive bad packet notification trap
if_retransmit ifTxRetransmit	Enables or disables OSPF interface retransmit notification trap
virt_retransmit virtIfTxRetransmit	Enables or disables OSPF virtual interface retransmit notification trap
orig_lsa originateLsa	Enables or disables OSPF new lsa origination notification trap
maxage_lsa maxAgeLsa	Enables or disables OSPF lsa maxage notification trap

syntax:

ospf [if_state < ifStateChange >] [virt_state < virtIfStateChange >] [nbr_state < nbrStateChange >] [vnbr_state < virtNbrStateChange >] [if_error < ifConfigError >] [virt_error < virtIfConfigError >] [if_auth < ifAuthFailure >] [virt_auth < virtIfAuthFailure >] [if_rxbad < ifRxBadpacket >] [virt_rxbad < virtIfRxBadpacket >] [if_retransmit < ifTxRetransmit >] [virt_retransmit < virtIfTxRetransmit >][orig_lsa < originateLsa >] [maxage_lsa < maxAgeLsa >]

example:

Router/configure/snmp-server> **enable_trap ospf ifAuthFailure**

related commands:

configure snmp-server enable traps bgp
configure snmp-server enable traps config
configure snmp-server enable traps environment
configure snmp-server enable traps failover
configure snmp-server enable traps frame_relay
configure snmp-server enable traps snmp
configure snmp-server enable traps sntp
configure snmp-server enable traps system
configure snmp-server enable traps vrrp

applicable models:

All models.

operating mode:

Routing only.

configure snmp-server enable traps snmp

This command enables or disables traps for SNMP authentication and failure.

parameter	definition
trap_auth	
auth_fail	Enable authentication failure trap

syntax:

snmp [trap_auth < auth_fail >]

example:

Router/configure/snmp/enable_trap> **snmp auth_fail**

related commands:

configure snmp-server enable traps bgp
configure snmp-server enable traps config
configure snmp-server enable traps environment
configure snmp-server enable traps failover
configure snmp-server enable traps frame_relay
configure snmp-server enable traps ospf
configure snmp-server enable traps sntp
configure snmp-server enable traps system
configure snmp-server enable traps vrrp

applicable models:

All models.

configure snmp-server enable traps sntp

This command enables or disables SNMP client traps for the simple network timing protocol.

syntax:

sntp

example:

Router/configure/snmp/enable_trap> **sntp**

related commands:

configure snmp-server enable traps bgp
configure snmp-server enable traps config
configure snmp-server enable traps environment
configure snmp-server enable traps failover
configure snmp-server enable traps frame_relay
configure snmp-server enable traps ospf
configure snmp-server enable traps sntp
configure snmp-server enable traps system
configure snmp-server enable traps vrrp

applicable models:

All models.

configure snmp-server enable traps system

This command enables or disables traps for reporting system shutdowns, user logins and logouts, and user login failures.

parameter	definition
trap_shutdown	
shutdown	Enables traps for Router system shutdowns.
trap_login	
logon	Enables traps for Router system logons.
trap_logoff	
logoff	Enables traps for Router system logoffs.
trap_loginfail	
loginfail	Enables traps for failed user login attempts.

syntax:

[no] system [trap_shutdown < shutdown >] [trap_login < logon >] [trap_logoff < logoff >] [trap_loginfail < loginfail >]

example:

Router/configure/snmp/enable_trap> **system shutdown**

related commands:

configure snmp-server enable traps bgp
configure snmp-server enable traps config
configure snmp-server enable traps environment
configure snmp-server enable traps failover
configure snmp-server enable traps frame_relay
configure snmp-server enable traps ospf
configure snmp-server enable traps snmp
configure snmp-server enable traps sntp
configure snmp-server enable traps vrrp

applicable models:

All models.

configure snmp-server enable traps vrrp

This command enables or disables VRRP group traps.

syntax:

vrrp

example:

Router/configure/snmp/enable_trap> vrrp

related commands:

configure snmp-server enable traps bgp
configure snmp-server enable traps config
configure snmp-server enable traps environment
configure snmp-server enable traps failover
configure snmp-server enable traps frame_relay
configure snmp-server enable traps ospf
configure snmp-server enable traps snmp
configure snmp-server enable traps sntp
configure snmp-server enable traps system

applicable models:

All models.

operating mode:

Routing only.

configure snmp-server snmp-source

This command configures the SNMP source IP address.

parameter	definition
snmp-source	The IP address of the SNMP source host.

syntax:

snmp-source <*IP address*>

example:

Router/configure/snmp-server> **source address 10.100.1.1**

related commands:

configure snmp-server community
configure snmp-server contact
configure snmp-server enable traps
configure snmp-server location
configure snmp-server REM
configure snmp-server REM_
configure snmp-server trap-source
configure snmp-server chassis-id

applicable models:

All models.

configure snmp-server trap-host

This command assigns IP addresses and names to the hosts that will receive SNMP traps from the Router system.

When configuring SNMP operation, use this command first to allow the system to send traps when it boots up and finds events to send as traps. Then, use the **configure snmp-server enable traps** command to specify which traps to report.

If multiple hosts are used, repeat this command for each host. Up to 10 hosts may be configured.

parameter	definition
trap-host	Host name of SNMP trap host. Enter an IP address.
community	SNMP community string. Enter a word; use up to 64 characters.

syntax:

trap_host host < *IP address* > community < *string* >

example:

Router/configure/snmp-server> **trap_host 100.10.10.4 Alarm_sys**

related commands:

configure snmp-server community
configure snmp-server contact
configure snmp-server enable traps
configure snmp-server location
configure snmp-server REM
configure snmp-server REM_
configure snmp-server trap-source
configure snmp-server chassis-id

applicable models:

All models.

configure snmp-server trap-source

This command configures SNMP trap messages so that they are sent to a specific IP address. Otherwise, traps will be sent to the system default address.

parameter	definition
host	IP address for the source of the SNMP trap messages.

syntax:

src_address host < *IP address* >

example:

Router/configure/snmp-server> **src_address 100.10.10.4**

related commands:

configure snmp-server community
configure snmp-server contact
configure snmp-server enable traps
configure snmp-server location
configure snmp-server REM
configure snmp-server REM_
configure snmp-server chassis-id
configure snmp-server trap-host

applicable models:

All models.

configure sntp

This command configures the simple network timing protocol for a specific server or client on the network.

parameter	definition
server	An IP address or a host (client) name The default is any broadcast server.
timeout	The time, in seconds, in which the host responds. The default is 1024 seconds.

syntax:

[no] sntp [server < *IP address* >] [timeout < n >]

example:

Router/configure> **sntp server 100.25.6.3**

applicable models:

All models.

configure ssh_keygen change

This command changes the passphrase used to encrypt a key file.

parameter	definition
oldpassphrase	The current passphrase.
newpassphrase	The new passphrase.
keyfile	The name of the encrypted key file.

syntax:

ssh_keygen change oldpassphrase <string> newpassphrase <string> keyfile <filename>

example:

Router/configure> **ssh_keygen change "oldphrase" "new phrase" testfile**

related commands:

configure ssh_keygen convert
configure ssh_keygen digest
configure ssh_keygen encrypt
configure ssh_keygen generate

applicable models:

All models.

configure ssh_keygen convert

This command converts a public key file in OpenSSH format (the default) to a Secure Shell Standard format, or vice versa.

parameter	definition
type	The type of conversion to perform, either secsh to convert from OpenSSH public key to SECSH public key or openssh to convert from unencrypted private or public SECSH key to OpenSSH.
keyfile	The name of the existing key file to convert.
newfile	The name of the new key file to create. If this parameter is omitted, the output is displayed on the screen

syntax:

ssh_keygen convert type <secsh | openssh> keyfile <filename> [newfile <filename>]

example:

Router/configure> **ssh_keygen convert secsh original.pub newfile new.pub**

related commands: .

configure ssh_keygen change
configure ssh_keygen digest
configure ssh_keygen encrypt
configure ssh_keygen generate

applicable models:

All models.

configure ssh_keygen digest

This command generates a public key digest of a key file. Used to compare key files.

parameter	definition
keyfile	The name of the key file.
digest	The type of digest to generate, either fingerprint of bubblebabble. The default is fingerprint

syntax:

ssh_keygen digest keyfile <*file_name*> [digest < fingerprint | bubblebabble>]

example:

Router/configure> **ssh_keygen digest tiara.pub fingerprint**

related commands: .

configure ssh_keygen change
configure ssh_keygen digest
configure ssh_keygen encrypt
configure ssh_keygen generate

applicable models:

All models.

configure ssh_keygen encrypt

This command encrypts a private key file. The private key used for secure shell server should not be passphrase protected because you must be able to start the secure shell server without manual intervention. This command can be used to encrypt the private key with keys unique to the Router router. The "no" form of the command can be used to decrypt the file.

parameter	definition
keyfile	The name of the key file to encrypt. The existing file is overwritten.

syntax:

ssh_keygen [no] encrypt keyfile <filename>

example:

Router/configure> **ssh_keygen encrypt testfile**

related commands:

configure ssh_keygen change
configure ssh_keygen digest
configure ssh_keygen generate

applicable models:

All models.

.

configure ssh_keygen generate

This command generates host and user authentication keys.

parameter	definition
type	The type of key to generate, either RSA or DSA.
outfile	The name of the files to contain the generated keys. The private key will be stored in a file with the provided file name, while the public key will be stored in a file with the same name with the extension ".pub". The default file name is shrsakey for RSA keys, and shdsakey for DSA keys.
passphrase	The passphrase to encrypt the key file. The default is a null string. This option should not be set when generating host keys.
bits	The length of the key, in bits. Valid values are from 512 to 2048. The default is 1024.
comment	A string to identify the key. The default is "user@hostname"

syntax:

ssh_keygen generate type <DSA | RSA> [outfile <filename>] [passphrase <phrase>] [bits <n>] [comment <comment>]

example:

Router/configure/ssh_keygen> **generate DSA comment "DSA host key"**

related commands: .

configure ssh_keygen change
configure ssh_keygen convert
configure ssh_keygen digest
configure ssh_keygen encrypt

applicable models:

All models.

configure ssh_server

This command initializes SSH server settings.

syntax:

ssh_server

example:

Router/configure> **ssh_server**

related commands:

configure ssh_server authentication
configure ssh_server cipher
configure ssh_server compression
configure ssh_server enable
configure ssh_server hostfile
configure ssh_server logevents
configure ssh_server mac
configure ssh_server port
configure ssh_server restore
configure ssh_server sftpd

applicable models:

All models.

configure ssh_server authentication

This command specifies whether or not the SSH server can use a specific user authentication method (password or public key). By default, both methods are enabled.

parameter	definition
method	The authentication method to be enabled or disabled.

syntax:

ssh_server [no] authentication method <password | publickey>

example:

Router/configure/ssh_server> **no authentication password**

related commands:

configure ssh_server cipher
configure ssh_server compression
configure ssh_server enable
configure ssh_server hostfile
configure ssh_server logevents
configure ssh_server mac
configure ssh_server port
configure ssh_server restore
configure ssh_server sftpd

applicable models:

All models.

configure ssh_server cipher

This command specifies whether or not the SSH server can use a specific cipher mechanism. Supported mechanisms include 3DES, Blowfish, and AES in CBC mode. By default, all cipher mechanisms are enabled.

parameter	definition
cipher	The cipher mechanism to enable or disable.

syntax:

ssh_server [no] cipher cipher <3descbc | blowfishcbc | aes128cbc | aes192cbc | aes256cbc>

example:

Router/configure/ssh_server> **no cipher blowfishcbc**

related commands: .

configure ssh_server authentication
configure ssh_server compression
configure ssh_server enable
configure ssh_server hostfile
configure ssh_server logevents
configure ssh_server mac
configure ssh_server port
configure ssh_server restore
configure ssh_server sftpd

applicable models:

All models.

configure ssh_server compression

This command specifies whether or not the SSH server can use a specific compression mechanism. By default, all compression mechanisms are enabled.

parameter	definition
mechanism	The type of compression being enabled or disabled.

syntax:

ssh_server [no] compression mechanism <none | zlib>

example:

Router/configure/ssh_server> **no compression zlib**

related commands:

configure ssh_server authentication
configure ssh_server cipher
configure ssh_server enable
configure ssh_server hostfile
configure ssh_server logevents
configure ssh_server mac
configure ssh_server port
configure ssh_server restore
configure ssh_server sftpd

applicable models:

All models.

configure ssh_server enable

This command enables or disables secure shell server.

syntax:

ssh_server [no] enable

example:

Router/configure/ssh_server> **enable**

related commands: r

configure ssh_server authentication
configure ssh_server cipher
configure ssh_server compression
configure ssh_server hostfile
configure ssh_server logevents
configure ssh_server mac
configure ssh_server port
configure ssh_server restore
configure ssh_server sftpd

applicable models:

All models.

configure ssh_server hostfile

This command specifies the name of the file containing the private host key for SSH operation. The default host key file is shdsakey. The server expects the public key to be stored in a file of the same name with the extension ".pub".

parameter	definition
hostfile	The name of the file containing the public host key.

syntax:

ssh_server hostfile hostfile <filename>

example:

Router/configure/ssh_server> **hostfile dsakey**

related commands:

configure ssh_server authentication
configure ssh_server cipher
configure ssh_server compression
configure ssh_server enable
configure ssh_server logevents
configure ssh_server mac
configure ssh_server port
configure ssh_server restore
configure ssh_server sftpd

applicable models:

All models.

configure ssh_server logevents

This command enables or disables SSH event logging. SSH logging is disabled by default.

syntax:

ssh_server [no] logevents

example:

Router/configure/ssh_server> **logevents**

related commands:

configure ssh_server authentication
configure ssh_server cipher
configure ssh_server compression
configure ssh_server enable
configure ssh_server hostfile
configure ssh_server mac
configure ssh_server port
configure ssh_server restore
configure ssh_server sftpd

applicable models:

All models.

configure ssh_server mac

This command specifies whether or not the SSH server can use a particular MAC algorithm.

By default, all algorithms are enabled.

parameter	definition
macs	The MAC algorithms.

syntax:

ssh_server [no] mac macs <hmacsha1 | hmacsha196 | hmacmd5 | hmacmd596>

example:

Router/configure/ssh_server> **no mac hmacmd5**

related commands:

configure ssh_server authentication
configure ssh_server cipher
configure ssh_server compression
configure ssh_server enable
configure ssh_server hostfile
configure ssh_server logevents
configure ssh_server port
configure ssh_server restore
configure ssh_server sftpd

applicable models:

All models.

configure ssh_server port

This command specifies the port on which the SSH server will listen.

parameter	definition
portno	The port number. The default is 22.

syntax:

ssh_server port portno <n>

example:

Router/configure/ssh_server> **port 2222**

related commands:

configure ssh_server authentication
configure ssh_server cipher
configure ssh_server compression
configure ssh_server enable
configure ssh_server hostfile
configure ssh_server logevents
configure ssh_server mac
configure ssh_server restore
configure ssh_server sftpd

applicable models:

All models.

configure ssh_server restore

This command restores the default values to all SSH configuration parameters.

syntax:

ssh_server restore

example:

Router/configure/ssh_server> **restore**

related commands:

configure ssh_server authentication
configure ssh_server cipher
configure ssh_server compression
configure ssh_server hostfile
configure ssh_server logevents
configure ssh_server mac
configure ssh_server port
configure ssh_server sftpd

applicable models:

All models.

configure ssh_server sftpd

This command enables or disables sftp server subsystem functionality, which is disabled by default.

syntax:

ssh_server [no] sftpd

example:

Router/configure/ssh_server> **sftpd**

related commands:

configure ssh_server authentication
configure ssh_server cipher
configure ssh_server compression
configure ssh_server enable
configure ssh_server hostfile
configure ssh_server logevents
configure ssh_server mac
configure ssh_server port
configure ssh_server restore

applicable models:

All models.

configure SYS_REM

This command allows users to add comments to the configuration file during a configuration session.

Comments will appear after using the **save local** command.

parameter	definition
comments	80-character (maximum) string enclosed in quotation marks.

syntax:

SYS_REM comments < "comments" >

example:

Router/configure> **SYS_REM "Configured on November 3, 2000. MNB 11/05/00 9:45."**

related commands:

display configuration stored

applicable models:

All models.

configure SYS_REM_

This command allows users to add comments (at the end of the configuration file) during a configuration session.

Comments will appear after using the **save local** command.

parameter	definition
comments	80-character (maximum) string enclosed in quotation marks.

syntax:

SYS_REM_ comments < "comments" >

example:

Router/configure> **SYS_REM_ "Everything left unchanged except IP filter. MNB 11/06/00 12:20."**

related commands:

display configuration stored

applicable models:

All models.

configure system

This command accesses next-level commands for configuring system-level functions.

syntax:

system

example:

Router/configure> **system**

next-level commands

configure system alarm_relay
configure system carrier_type
configure system licenses
configure system logging
configure system mac_range
configure system routing

applicable models:

All models.

configure system alarm_relay

This command configures the state of the alarm relay for non-alarm reporting.

parameter	definition
state	
open	Relay contacts are open when no Summary alarm is active
closed	Relay contacts are closed when no Summary alarm is active This is the default value.

syntax:

alarm_relay state < open | closed >

example:

Router/configure/system> **alarm_relay open**

related commands:

configure system carrier_type
configure system licenses
configure system logging
configure system mac_range
configure system routing

applicable models:

All models, except OmniAccess 604.

configure system carrier_type

This command configures the type of carrier being used. Select either T1 or E1. The router must reboot before the carrier type change takes effect.

parameter	definition
0	Specifies a T1 carrier.
1	Specifies an E1 carrier.

syntax:

carrier_type < 0 | 1 >

example:

Router/configure/system> **carrier_type 1**

related commands:

configure system licenses
configure system logging
configure system mac_range
configure system routing

applicable models:

OmniAccess 601

configure system hdlc_error

This command configures the threshold for the number of consecutive HDLC errors received.

parameter	definition
hdlc_error	The number of consecutive HDLC errors. The default is 1000. Valid range is 100 to 12000.

syntax:

hdlc_error

example:

Router/configure/system> **hdlc_error 250**

related commands:

configure system hdlc_link_deactivate

applicable models:

All models.

configure system hdlc_link_deactivate

This command is a system-wide setting which deactivates links when the excessive HDLC error threshold is exceeded.

syntax:

hdlc_link_deactivate

example:

Router/configure/system> **hdlc_link_deactivate**

related commands:

configure interface bundle hdlc_link_activate

applicable models:

All models, except the OmniAccess 601.

configure system licenses

This command configures feature upgrade licenses and/or enables/disables routing.

parameter	definition
license_type	Specifies the type of feature upgrade license

syntax:

licenses license_type <port_enable> | < routing >

example 1:

Router/configure/system> **licenses vpn_advance**

NOTE: You will be prompted to enter an 18-character upgrade license key to enable this feature. Obtain this key from Router Customer Support.

related commands:

configure system alarm_relay
configure system carrier_type
configure system logging
configure system mac_range
configure system routing
show system licenses

applicable models:

All models.

configure system logging

This command accesses next-level commands for configuring console logging and syslog event reporting.

syntax:

logging

example:

Router/configure/system> **logging**

next-level commands

configure system logging console
configure system logging syslog

applicable models:

All models.

configure system logging console

This command configures system messages to be sent to the console.

They are defined hierarchically by severity, with "emergency" messages being the most urgent and "debugging" messages being the least.

Users have the option of selecting which message types they wish to have sent to the console. All messages of the selected message type (and messages of greater severity) will be sent to the console.

parameter	definition
level	
emergency	Emergency messages are logged.
alert	Alert messages are logged, plus emergency messages.
critical	Critical messages are logged, plus all messages described above. This is the default.
error	Error messages are logged, plus all messages described above.
warning	Warning messages are logged, plus all messages described above.
notification	Notification messages are logged, plus all messages described above.
informational	Informational messages are logged, plus all messages described above.
debugging	Debugging messages are logged, plus all messages described above.

syntax:

console [level < emergency | alert | critical | error | warning | notification | informational | debugging >]

example:

Router/configure/system/logging> **console critical**

related commands:

configure system logging syslog

applicable models:

All models.

configure system logging syslog

This command accesses next-level commands for configuring syslog on the Router system.

syntax:

syslog

example:

Router/configure/system/logging> **syslog**

next-level commandsc

configure system logging syslog auth
configure system logging syslog bootp
configure system logging syslog bundle
configure system logging syslog daemon
configure system logging syslog domainname
configure system logging syslog enable (The default value.)
configure system logging syslog facility
configure system logging syslog fr
configure system logging syslog gated
configure system logging syslog hdlc
configure system logging syslog host_ipaddr
configure system logging syslog ipmux
configure system logging syslog kern
configure system logging syslog local7
configure system logging syslog mail
configure system logging syslog ntp
configure system logging syslog ppp
configure system logging syslog qos
configure system logging syslog system
configure system logging syslog vpn

applicable models:

All models.

configure system logging syslog auth

This command sets the authorization facility logging message priority.

The priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "debugging" messages being the least.

parameter	definition
message type	
none	Does not log messages of the specified facility.
emerg	Emergency messages are logged.
alert	Alert messages are logged, plus emergency messages.
crit	Critical messages are logged, plus all messages described above.
err	Error messages are logged, plus all messages described above.
warning	Warning messages are logged, plus all messages described above (default).
notice	Notification messages are logged, plus all messages described above.
info	Informational messages are logged, plus all messages described above.
debug	Debugging messages are logged, plus all messages described above.

syntax:

auth message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **auth emerg**

applicable models:

All models.

configure system logging syslog bootp

This command sets the BootP facility logging message priority.

The priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log messages of the specified facility.
emerg	Emergency messages are logged.
alert	Alert messages are logged, plus emergency messages.
crit	Critical messages are logged, plus all messages described above.
err	Error messages are logged, plus all messages described above.
warning	Warning messages are logged, plus all messages described above (default).
notice	Notification messages are logged, plus all messages described above.
info	Informational messages are logged, plus all messages described above.
debug	Debugging messages are logged, plus all messages described above.

syntax:

bootp message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **bootp alert**

applicable models:

All models.

configure system logging syslog bundle

This command sets the bundle facility logging message priority.

The priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log messages of the specified facility.
emerg	Emergency messages are logged.
alert	Alert messages are logged, plus emergency messages.
crit	Critical messages are logged, plus all messages described above.
err	Error messages are logged, plus all messages described above.
warning	Warning messages are logged, plus all messages described above (default).
notice	Notification messages are logged, plus all messages described above.
info	Informational messages are logged, plus all messages described above.
debug	Debugging messages are logged, plus all messages described above.

syntax:

bundle message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **bundle crit**

applicable models:

All models.

configure system logging syslog daemon

This command configures daemon facility logging message priority.

Priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log daemon messages
emerg	Logs only daemon emergency messages
alert	Logs only daemon alert and above messages
crit	Logs only daemon critical and above messages
err	Logs only daemon error and above messages
warning	Logs only daemon warning and above messages This is the default configuration.
notice	Logs only daemon notification and above messages
info	Logs only daemon informational and above messages
debug	Logs all daemon messages

syntax:

daemon message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **daemon err**

applicable models:

All models.

configure system logging syslog domainname

This command configures domain name system facility logging message priority.

Priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log domainname messages
emerg	Logs only domainname emergency messages
alert	Logs only domainname alert and above messages
crit	Logs only domainname critical and above messages
err	Logs only domainname error and above messages
warning	Logs only domainname warning and above messages This is the default configuration.
notice	Logs only domainname notification and above messages
info	Logs only domainname informational and above messages
debug	Logs all domainname messages

syntax:

domainname message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **domainname warning**

applicable models:

All models.

configure system logging syslog enable

This command enables or disables syslog.

parameter	definition
enable	Enables syslog

syntax:

[no] enable

example:

Router/configure/system/logging/syslog> **enable**

applicable models:

All models.

configure system logging syslog facility

This command configures the Cisco-style facility type.

parameter	definition
facility	
sys9	Indicates system use
sys10	Indicates system use
sys11	Indicates system use
sys12	Indicates system use
sys13	Indicates system use
sys14	Indicates system use
local0	Locally defined messages
local1	Locally defined messages
local2	Locally defined messages
local3	Locally defined messages
local4	Locally defined messages
local5	Locally defined messages
local6	Locally defined messages
local7	Locally defined messages

syntax:

facility facility < sys 9 | sys10 | sys11 | sys12 | sys13 | sys14 | local0 | local1 | local2 | local3 | local 4 | local5 | local6 | local7 >

example:

Router/configure/system/logging/syslog> **facility local0**

applicable models:

All models.

configure system logging syslog fr

This command sets the frame relay facility logging message priority.

The priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log any messages.
emerg	Emergency messages are logged.
alert	Alert messages are logged, plus emergency messages.
crit	Critical messages are logged, plus all messages described above.
err	Error messages are logged, plus all messages described above.
warning	Warning messages are logged, plus all messages described above. This is the default.
notice	Notification messages are logged, plus all messages described above.
infor	Informational messages are logged, plus all messages described above.
debug	Debugging messages are logged, plus all messages described above.

syntax:

fr message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **fr notice**

applicable models:

All models.

configure system logging syslog gated

This command configures gateway facility logging message priority.

Priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log gated messages
emerg	Logs only gated emergency messages
alert	Logs only gated alert and above messages
crit	Logs only gated critical and above messages
err	Logs only gated error and above messages
warning	Logs only gated warning and above messages This is the default configuration.
notice	Logs only gated notification and above messages
info	Logs only gated informational and above messages
debug	Logs all gated messages

syntax:

gated message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **gated info**

applicable models:

All models.

configure system logging syslog hdlc

This command sets the HDLC facility logging message priority.

The priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
messsage type	
none	Does not log any messages.
emerg	Emergency messages are logged.
alert	Alert messages are logged, plus emergency messages.
crit	Critical messages are logged, plus all messages described above.
err	Error messages are logged, plus all messages described above.
warning	Warning messages are logged, plus all messages described above. This is the default.
notice	Notification messages are logged, plus all messages described above.
infor	Informational messages are logged, plus all messages described above.
debug	Debugging messages are logged, plus all messages described above.

related commands:

hdlc message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **hdlc info**

applicable models:

All models.

configure system logging syslog host_ipaddr

This command sets the syslog host IP address.

parameter	definition
host_ipaddr	Syslog host IP address
udp_portno	Syslog host UDP port number The range is 1 - 65535; the default is 514.

syntax:

host_ipaddr host_ipaddr < *IP address* > [udp_portno < n >]

example:

Router/configure/system/logging/syslog> **host_ipaddr 10.1.3.9**

applicable models:

All models.

configure system logging syslog ipmux

This command sets the IP multiplexing facility logging message priority.

The priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log any messages.
emerg	Emergency messages are logged.
alert	Alert messages are logged, plus emergency messages.
crit	Critical messages are logged, plus all messages described above.
err	Error messages are logged, plus all messages described above.
warning	Warning messages are logged, plus all messages described above. This is the default.
notice	Notification messages are logged, plus all messages described above.
infor	Informational messages are logged, plus all messages described above.
debug	Debugging messages are logged, plus all messages described above.

syntax:

ipmux message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **ipmux none**

applicable models:

All models.

configure system logging syslog kern

This command configures the kernal facility logging message priority.

Priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log kernal messages
emerg	Logs only kernal emergency messages
alert	Logs only kernal alert and above messages
crit	Logs only kernal critical and above messages
err	Logs only kernal error and above messages This is the default configuration value.
warning	Logs only kernal warning and above messages
notice	Logs only kernal notification and above messages
info	Logs only kernal informational and above messages
debug	Logs all kernal messages

syntax:

kern message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **kern err**

applicable models:

All models.

configure system logging syslog local7

This command configures user defined facility logging message priority.

Priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log user-defined messages
emerg	Logs only user-defined emergency messages
alert	Logs only user-defined alert and above messages
crit	Logs only user-defined critical and above messages
err	Logs only user-defined error and above messages
warning	Logs only user-defined warning and above messages This is the default configuration.
notice	Logs only user-defined notification and above messages
info	Logs only user-defined informational and above messages
debug	Logs all user-defined messages

syntax:

local7 message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **local7 emerg**

applicable models:

All models.

configure system logging syslog mail

This command configures mail utility facility logging message priority.

Priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log mail messages
emerg	Logs only mail emergency messages
alert	Logs only mail alert and above messages
crit	Logs only mail critical and above messages
err	Logs only mail error and above messages
warning	Logs only mail warning and above messages This is the default configuration.
notice	Logs only mail notification and above messages
info	Logs only mail informational and above messages
debug	Logs all mail messages

syntax:

mail message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **mail none**

applicable models:

All models.

configure system logging syslog ntp

This command configures network time protocol facility logging message priority.

Priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log ntp messages
emerg	Logs only ntp emergency messages
alert	Logs only ntp alert and above messages
crit	Logs only ntp critical and above messages
err	Logs only ntp error and above messages
warning	Logs only ntp warning and above messages This is the default configuration.
notice	Logs only ntp notification and above messages
info	Logs only ntp informational and above messages
debug	Logs all ntp messages

syntax:

ntp message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **ntp debug**

applicable models:

All models.

configure system logging syslog ppp

This command sets the PPP facility logging message priority.

The priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log any messages.
emerg	Emergency messages are logged.
alert	Alert messages are logged, plus emergency messages.
crit	Critical messages are logged, plus all messages described above.
err	Error messages are logged, plus all messages described above.
warning	Warning messages are logged, plus all messages described above. This is the default.
notice	Notification messages are logged, plus all messages described above.
infor	Informational messages are logged, plus all messages described above.
debug	Debugging messages are logged, plus all messages described above.

syntax:

ppp message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **ppp warning**

applicable models:

All models.

configure system logging syslog qos

This command sets the QoS facility logging message priority.

The priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log messages of the specified facility.
emerg	Emergency messages are logged.
alert	Alert messages are logged, plus emergency messages.
crit	Critical messages are logged, plus all messages described above.
err	Error messages are logged, plus all messages described above.
warning	Warning messages are logged, plus all messages described above (default).
notice	Notification messages are logged, plus all messages described above.
info	Informational messages are logged, plus all messages described above.
debug	Debugging messages are logged, plus all messages described above.

syntax:

qos message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **qos notice**

applicable models:

All models.

configure system logging syslog system

This command sets the system facility logging message priority.

The priorities are defined hierarchically by severity, with "emergency" messages being the most urgent and "informational" messages being the least.

parameter	definition
message type	
none	Does not log messages of the specified facility.
emerg	Emergency messages are logged.
alert	Alert messages are logged, plus emergency messages.
crit	Critical messages are logged, plus all messages described above.
err	Error messages are logged, plus all messages described above.
warning	Warning messages are logged, plus all messages described above (default).
notice	Notification messages are logged, plus all messages described above.
info	Informational messages are logged, plus all messages described above.
debug	Debugging messages are logged, plus all messages described above.

syntax:

system message type < none | emerg | alert | crit | err | warning | notice | info | debug >

example:

Router/configure/system/logging/syslog> **system info**

applicable models:

All models.

configure system mac_range

This command distributes the available MAC addresses between the Ethernet ports.

A factory default IP address is assigned to Ethernet 1. This default IP address should be changed when the Router system is initially configured.

parameter	definition
ethernet	Number of the Ethernet port to be configured (0 or 1).
range	The range of MAC addresses to be distributed on a specific Ethernet port The range is 5 - 1024.

syntax:

mac_range ethernet < 0 | 1 > range < n >

example:

Router/configure/system> **mac_range 1 80**

related commands:

configure system alarm_relay
configure system carrier_type
configure system licenses
configure system logging
configure system routing

applicable models:

All models.

NOTE: Be sure to designate at least one MAC address per Ethernet port for management purposes.

configure system reset-to-factory

This command returns the router to factory default settings.

syntax:

reset-to-factory

example:

Router/configure/system> **reset-to-factory**

related commands:

configure system carrier_type
configure system licenses
configure system logging
configure system mac_range
configure system routing

applicable models:

All models.

configure system routing

This command configures dynamic routing.

Before enabling dynamic routing, use the **configure system licenses routing** command to enable the Router system to execute routing protocol commands. Executing this command requires entering an 18-character routing license key, which can be obtained from Router Technical Support.

syntax:

routing

example:

Router/configure/system> **routing**

If the routing license key has not been previously entered, you will be prompted with the following message:

"You must have a valid routing upgrade license key to enable dynamic routing mode!"

If the routing license key has been installed, the system will reboot after issuing the **configure system routing** command. To preserve your configuration settings, use the **save local command** prior to issuing the **configure system routing** command.

related commands:

configure system alarm_relay
configure system carrier_type
configure system licenses
configure system logging
configure system mac_range

applicable models:

All models.

configure telnet_banner

This command is used to configure a message to be displayed after a telnet login.

To configure a message of more than 80 characters, issue the **telnet_banner1** command (for the second line of text). To add a third text line to the message, use the **telnet_banner2** command. The entire message cannot exceed 255 characters. To insert a blank line between text lines, type"\n" before the text of the line following the blank line.

parameter	definition
banner	Enter up to 80 characters in a quoted string.

syntax:

[no] telnet_banner banner < "*banner*" >

example:

Router/configure> **telnet_banner "This system is private property and is only for the use of authorized personnel."**

applicable models:

All models.

configure telnet_timeout

This command sets the time, in seconds, that a Telnet session will automatically end.

parameter	definition
timeout	The time, in seconds The range is 0 - 3600 seconds; the default is 900 seconds. If 0 is entered, there will be no timeout configured.

syntax:

telnet_timeout timeout < n >

example:

Router/configure> **telnet_timeout 1300**

applicable models:

All models.

configure terminal

From a workstation or Telnet session connected to a Router system, **configure terminal** accesses all next-level **configure** commands for setting all system parameters.

syntax:

configure terminal

example:

Router> **configure terminal**

next-level commands

configure terminal monitor

related commands:

configure flash
configure network

applicable models:

All models.

configure terminal monitor

This command enables or disables the display of debug information at a remote monitor that is accessing the network over a telnet connection.

syntax:

[no] monitor

example:

Router/configure/terminal> **monitor**

applicable models:

All models.

configure tftp_server

This command enables (disables) the TFTP server.

syntax:

[no] tftp_server

example:

Router/configure/terminal> **tftp_server**

applicable models:

All models.

configure user

This command adds or removes system users.

You can configure and assign access privilege levels for up to 15 users. To remove a user, type **no user name**, followed by that user's name.

parameter	definition
name	Name of the user (up to 30 alphanumeric characters).
level	User access privilege level, as follows:
1	System Administrator; Access to all CLI operations.
2	Can configure the system, view system data, conduct tests, and change the user's current password. However, this user cannot add users to or remove users from the system.
3	Can view system data, conduct tests, and change user's current password. However, this user cannot perform any other operations.
4	Can view system data and change user's current access password, but cannot perform any other operations. This level is automatically assigned if a level is not specified.
	Default value.

syntax:

[no] user name < name > [level < level > < 1 | 2 | 3 | 4 >]

example 1:

Router/configure> **user name John level 2**

This example adds a user named John with level 2 access privileges.

example 2:

Router/configure> **no user name John**

This example removes a user named John.

applicable models:

All models.

NOTE: Only the System Administrator (Level 1 access) can add users to and remove users from a Router system. System Administrators will also be prompted to enter a temporary password for each user that they add.

configure utc

This command sets the current time for the router in Universal Time Coordinated time which is Earth's official time, also known as Greenwich Mean Time. Also referred to as Zulu time. Time is expressed in terms of a 24-hour clock (00:00:00 to 23:59:59 hours).

All global timezones are identified by their offset from Greenwich Mean Time - with timezones west of Greenwich England signed as negative as they are behind, and timezones east of Greenwich signed as positive as they are ahead of Greenwich time.

parameter	definition
time_zone_offset	Enter a plus sign (+) for timezones ahead of Greenwich - that is, places where the local time is earlier than it is in Greenwich England. Enter a minus sign (-) for timezones behind Greenwich - that is places where the local time is later than it is in Greenwich England.
UTC hours	The number of hours ahead or behind Greenwich. Valid range is 00 - 23. All time is expressed in terms of a 24-hour clock.
UTC minutes	The number of minutes ahead or behind Greenwich. Valid range is 00 - 59. (Most timezones only need to set the number of hours of offset.)

syntax:

[no] utc time_zone_offset [UTCHours] [UTCMinutes]

example:

To set the local Pacific Standard Time to UTC time, enter:

Router_LA/configure> **utc - UTCHours 8**

Router_LA/configure> **show date**

Local Date and Time:

MON NOV 08 11:15:05 2004

UTC Date and Time:

MON NOV 08 19:15:05 2004

UTC time_zone_offset: -8:0

related commands:

configure date

applicable models:

All models.

configure vlanfwd

This command accesses next-level commands for configuring VLAN forwarding.

syntax:

[no] vlanfwd

example:

Router/configure> **vlanfwd**

next-level commands

configure vlanfwd add
configure vlanfwd enable
configure vlanfwd ether_type
configure vlanfwd management
configure vlanfwd REM
configure vlanfwd REM_

applicable models:

All models.

configure vlanfwd add

This command accesses next-level commands for configuring a VLAN.

syntax:

[no] add

example:

Router/configure vlanfwd> **add**

next-level commands

configure vlanfwd add vlanid

applicable models:

All models.

configure vlanfwd add vlanid

This command configures a VLAN.

parameter	definition
vlan_id	VLAN id. The range is 1 - 4095, single entry (100) or range (200 - 300)
interface	Interface name Enter ethernet0, ethernet1, or a bundle name.

syntax:

[no] add vlanid vlan_id < n > interface < ethernet0 | ethernet1 | bundle name | bundle name:PVC number >

example:

Router/configure/vlanfwd> **add vlanid 120 ethernet0**

applicable models:

All models.

configure vlanfwd add vldid

This command creates or deletes a VLAN domain (VLD) entry in the VLD forwarding table.

parameter	definition
vlan_id	VLAN id. The range is 1 - 4095, single entry (100) or range (200 - 300)
interface	Interface name Enter ethernet0, ethernet1, or a bundle name.

syntax:

[no] add vldid vld_id < n > interface < ethernet0 | ethernet1 | bundle_name >

example:

host/configure/vlanfwd> **add vldid 120 ethernet0**

applicable models:

All models.

configure vlanfwd enable

This command enables VLAN forwarding. The no option of this command temporarily disables VLAN forwarding.

By default ,VLAN forwarding is enabled when vlan forwarding is configured (when the user enters the vlanfwd tree). The "no" option of this command temporarily prevents the forwarding of VLAN packets, but it does not remove any VLAN configuration. Thus, the user is free to continue configuration without forwarding VLAN packets. To reset VLAN forwarding, execute the **configure vlanfwd enable** command.

Also, when a **save local** command is executed after a **no vlan enable** command, the VLAN forwarding table and associated configuration will not be saved.

syntax:

[no] enable

example 1:

Router/configure/vlanfwd> **enable**

example 2:

To enable Layer-3 translation for VLAN frames, enter:

Router/configure/vlanfwd/layer3/translation> **enable**

related commands:

configure vlanfwd add
configure vlanfwd ether_type
configure vlanfwd management
configure vlanfwd REM
configure vlanfwd REM_

applicable models:

All models.

configure vlanfwd ether_type

This command configures the Ethernet type for VLAN forwarding.

parameter	definition
ether_type	Ethernet type, in decimal format The range is 1 - 65535; the default is 33024 (0 x 8100).

syntax:

ether_type [ether_type < n >]

example:

Router/configure/vlanfwd> **ether_type 33027**

related commands:

configure vlanfwd add
configure vlanfwd enable
configure vlanfwd management
configure vlanfwd REM
configure vlanfwd REM_

applicable models:

All models.

configure vlanfwd macbridge

This command enables or disables VLAN bridging.

When bridging is enabled, this command accesses next-level commands. Bridging is enabled by default.

syntax:

[no] macbridge

example:

Router/configure/vlanfwd> **macbridge**

next-level commands

configure vlanfwd macbridge add
configure vlanfwd macbridge age
configure vlanfwd macbridge enable

applicable models:

All models.

configure vlanfwd macbridge add

This command accesses next-level commands.

syntax:

add

example:

Router/configure/vlanfwd/macbridge> **add**

next-level commands

configure vlanfwd macbridge add macentry

applicable models:

All models.

configure vlanfwd macbridge add macentry

This command adds or deletes a static MAC address entry to the forwarding database.

parameter	definition
macentry	MAC address to be added or deleted

syntax:

[no] add macentry < macaddress > < interface name >

example:

Router/configure/vlanfwd/macbridge> **add macentry 00:50:52:02:04:06 ethernet0**

applicable models:

All models.

configure vlanfwd macbridge age

This command configures the age of dynamically learned MAC addresses.

The new value will take effect on future new entries or on existing entries as they are refreshed. Otherwise, existing database entries are not changed.

parameter	definition
age	The period (in minutes) for which dynamically learned MAC address entries are valid. The range is 1 - 1440; the default is 5 minutes.

syntax:

[no] age age < n >

example:

Router/configure/vlanfwd/macbridge> **age 10**

related commands:

configure vlanfwd macbridge add
configure vlanfwd macbridge enable

applicable models:

All models.

configure vlanfwd macbridge enable

This command enables or disables VLAN bridging.

Using the no form of this command disables macbriging and allows the user to continue to add static MAC entries and reconfigure the aging parameter. The **configure vlanfwd macbridge** command has no effect once bridging has been disabled with the **no enable** command. To return to bridging mode, use the **enable** command.

syntax:

[no] enable

example:

Router/configure/vlanfwd/macbridge> **enable**

related commands:

configure vlanfwd macbridge add
configure vlanfwd macbridge age

applicable models:

All models.

configure vlanfwd management

This command accesses next-level commands for configuring VLAN forwarding management.

These commands allow users to enable in-band VLAN management: add a destination host to the VLAN management table, set the time for VLAN management entries, add default routes, enable and disable forwarding of untagged packets, and configure the VLAN management ID.

syntax:

management

example:

Router/configure/vlanfwd> **management**

next-level commands

configure vlanfwd management add_host
configure vlanfwd management aging_interval
configure vlanfwd management default_route
configure vlanfwd management disable_ipfwd
configure vlanfwd management ip_interface
configure vlanfwd management vlanid

applicable models:

All models.

configure vlanfwd management add_host

This command adds a destination host to the VLAN management table.

parameter	definition
ip_address	The host IP address
interface	The outgoing interface name Enter a bundle name or an Ethernet port (1 or 2).
mac_address	The host MAC address in the form aa:bb:cc:dd:ee:ff

syntax:

add_host ip_address < *IP address* > interface < ethernet0 | ethernet1 | bundle name | bundle name:PVC number > mac_address < *MAC address* >

example:

Router/configure/vlanfwd/management> **add_host 101.12.2.1 network_01 11:22:33:44:55:66**

related commands:

configure vlanfwd management aging_interval
configure vlanfwd management default_route
configure vlanfwd management disable_ipfwd
configure vlanfwd management ip_interface
configure vlanfwd management vlanid

applicable models:

All models.

configure vlanfwd management aging_interval

This command configures the expiration time (in minutes) for dynamic VLAN entries.

After the time expires, the Router system will drop a particular VLAN entry (its MAC address) from the management table.

parameter	definition
aging_interval	The expiration time for VLAN management entries in minutes The range is 1 - 65535 minutes; the default is 20 minutes.

syntax:

aging_interval [aging_interval < n >]

example:

Router/configure/vlanfwd/management> **aging_interval 30**

related commands:

configure vlanfwd management add_host
configure vlanfwd management default_route
configure vlanfwd management disable_ipfwd
configure vlanfwd management ip_interface
configure vlanfwd management vlanid

applicable models:

All models.

configure vlanfwd management default_route

This command configures a default route (within a VLAN network itself) where off-network traffic is forwarded.

parameter	definition
gateway	A gateway IP address
interface	The outgoing interface name - ethernet0, ethernet1, or bundle name
mac_address	The gateway MAC address in the form of aa:bb:cc:dd:ee:ff

syntax:

[no] default_route gateway < *IP address* > interface < ethernet0 | ethernet1 | bundle name | bundle name:PVC number > [mac_address < *MAC address* >]

example:

Router/configure/vlanfwd/management> **default_route 101.12.2.2 network_02 11:22:33:44:55:66**

related commands:

configure vlanfwd management add_host
configure vlanfwd management aging_interval
configure vlanfwd management disable_ipfwd
configure vlanfwd management ip_interface
configure vlanfwd management vlanid

applicable models:

All models.

NOTE: Both the gateway and interface must be on the same subnet.

configure vlanfwd management disable_ipfwd

This command enables or disables the forwarding of untagged IP packets.

syntax:

[no] disable_ipfwd

example:

Router/configure/vlanfwd/management> **no disable_ipfwd**

related commands:

configure vlanfwd management add_host
configure vlanfwd management aging_interval
configure vlanfwd management default_route
configure vlanfwd management ip_interface
configure vlanfwd management vlanid

applicable models:

All models.

configure vlanfwd management ip_interface

This command creates the VMI interface named VlanMgt and accesses next-level commands.

syntax:

ip_interface

example:

Router/configure/vlanfwd/management> **ip_interface**

next-level commands

configure vlanfwd management ip_interface address

applicable models:

All models.

configure vlanfwd management ip_interface address

This command configures the IP address and subnet mask to create a VLAN management interface named VlanMgt.

This VMI interface creates a loopback interface that is used to manage the Router router in VLAN management mode. The IP address should be in the same subnet as the VLAN management domain. Use the **default_route** command with this VMI interface to specify the outbound interface for off-network traffic.

syntax:

address ip address < ip address > net mask < netmask >

example:

Router/configure/vlanfwd/management/ip_interface> **address 10.1.1.1 24**

applicable models:

All models.

configure vlanfwd management vlanid

This command configures the VLAN management ID.

parameter	definition
vlanid	The VLAN management ID number. The range is 1 - 4095; the default is 4092.

syntax:

[no] vlanid vlanid < n >

NOTE: Using the “no” option of this command deletes the specified vlanid and sets the vlanid to the default value, 4092.

example:

Router/configure/vlanfwd/management> **vlanid 300**

related commands:

configure vlanfwd management add_host
configure vlanfwd management aging_interval
configure vlanfwd management disable_ipfwd
configure vlanfwd management default_route
configure vlanfwd management ip_interface

applicable models:

All models.

configure vlanfwd REM

This command allows users to add comments (at the beginning of the VLAN area of the configuration file) during a configuration session.

Comments will appear after using the **save local** command.

parameter	definition
comments	80-character (maximum) string enclosed in quotation marks.

syntax:

REM comments < "comments" >

example:

Router/configure/vlanfwd> **REM "VLAN forwarding configured on November 5, 2000.TVD 11:50."**

related commands:

display configuration stored

applicable models:

All models.

configure vlanfwd REM_

This command allows users to add comments (at the end of the VLAN area of the configuration file) during a configuration session.

Comments will appear after using the **write memory** command.

parameter	definition
comments	80-character (maximum) string enclosed in quotation marks.

syntax:

REM_ comments < "comments" >

example:

host/configure/vlanfwd> **REM_ "VLAN forwarding configured on November 5, 2000.TVD 11:50."**

related commands:

display configuration stored

applicable models:

All models.

configure vlanfwd vld_ether_type

This command globally configures the vld Ethernet type for the system.

Unless a specific vld_ether_type is configured on an interface, this value is inherited.

parameter	definition
vld_ether_type	Ethernet type in decimal form The range is 1 - 65535; the default is 33024 (0x8100)

syntax:

[no] vld_ether_type < n | n - n >

example 1:

Set Ethernet type to 41216 (0xa100) for VLAN domain frames.

Router/configure/vlanfwd> **vld_ether_type 41216**

example 2:

Set the Ethernet type to default for VLAN domain frames.

Router/configure/vlanfwd> **no vld_ether_type 41216**

applicable models:

All models.

police rate

This command allows users to rate limit inbound traffic on the WAN links using policing while doing Class-Based Queueing (CBQ) for outbound traffic. This means the router can now provide QoS for traffic in both directions, eliminating dependency on the upstream router. Additionally, CBQ provides a bandwidth guarantee, bandwidth borrowing, and prioritization. Traffic policing is also supported on Ethernet interfaces.

Traffic policing is implemented using a token bucket algorithm. Users will be able specify two parameters when configuring traffic policing, Rate (token fill rate) and Burst (number of tokens). Packets conforming to these limits will be forwarded, and those violating these limits will be dropped. (Other conform-actions and violate-actions, like marking down the ToS or DSCP value, are not supported in this release.)

To disable traffic policing, use the **no police** command.

parameter	definition
rate	The token fill rate in Kbps. Determines the average bandwidth for the policed flow.
burst	Number of tokens specified in kilobits. Determines the maximum burst (in bits or bytes) permitted for the flow.
burst-time	Duration (based on the configured rate) in milliseconds.

syntax:

police rate <rate> [burst <rate> | burst-time <time>]

example:

Class c1 is configured for policing with a rate of 512 Kbps and a burst of 768 Kbits. This translates to a permissible burst of 1.5 sec at the rate of 512 Kbps.

R1/configure/interface/bundle wan1/qos/class c1> **police rate 512 burst 768**

R1/configure/interface/bundle wan1/qos/class c1> **exit**

R1/configure/interface/bundle wan1/qos> **class c2**

R1/configure/interface/bundle wan1/qos/class c2> **police rate 1024 burst-time 800**

R1/configure/interface/bundle wan1/qos/class c2> **exit**

R1/configure/interface/bundle wan1/qos> **class c3**

R1/configure/interface/bundle wan1/qos/class c3> **police rate 256**

related commands:

display configuration stored

applicable models:

All models.

4
DEBUG

The Router operating system includes a comprehensive suite of debug commands for use by network administrators for network configuration and troubleshooting.

NOTE: Debug commands are resource intensive. Because certain debug commands (such as **arp**) run in the background, they consume system resources. Using these commands will impact system performance, therefore they should be used with discretion. And, it is not advisable to run debug commands on a system that is operating in a live network.

The first-level debug commands are as follows:

Top-Level Debug Commands

crypto	Accesses debug crypto commands.
dhcp_relay	Accesses dhcp relay debug commands.
fr	Accesses frame relay debug commands.
ip	Accesses IP/IPmux debug commands.
lance	Accesses Ethernet debug commands.
mhu	Enables or disables Multi-Hospitality Unit (MHU) debug commands.
nat	Enables or disables Network Access Translation (NAT) debug commands.
ppp	Enables or disables Point-to-Point Protocol debug commands.
qos	Enables or disables Quality of Service (QoS) debug commands.
rtp	Displays debug information about Real-time Transport Protocol (RTP).
system	Accesses system-level debug commands.
ttcp	Accesses commands to configure the ttcp server and client.
vlan	Accesses Virtual Local Area Network (VLAN) debug commands.

debug all

debug all

This command disables all current debugging.

syntax:

no debug all

example:

Router1> **no debug all**

applicable models:

All models.

debug crypto

This command accesses next-level crypto debug commands

syntax:

crypto

example:

Router1/debug> **crypto**

next-level commands

debug crypto all
debug crypto ike
debug crypto ipsec

applicable models:

All models.

debug crypto all

This command enables or disables all IPSec and firewall debugging.

syntax:

[no] all

example:

Router1/debug/crypto> **all**

related commands:

debug crypto ike
debug crypto ipsec

applicable models:

All models.

debug crypto ike

This command enables or disables IKE debugging.

syntax:

[no] ike

example:

Router1/debug/crypto> **ike**

related commands:

debug crypto all
debug crypto ipsec

applicable models:

All models.

debug crypto ipsec

This command enables or disables IPSec debugging.

parameter	definition
mode	
ipsec	IPSec debugging This is the default
spd	SPD debugging
all	All IPSec debugging

syntax:

[no] ipsec [mode < ipsec | spd | all >]

example:

Router1/debug/crypto> **ipsec spd**

related commands:

debug crypto all
debug crypto ike

applicable models:

All models.

debug dhcp_relay

This command accesses next-level dhcp relay debug commands.

syntax:

dhcp_relay

example:

Router/dubug> **dhcp_relay**

next-level commands

debug dhcp_relay display_dhcp_table
debug dhcp_relay display_hash_statistics
debug dhcp_relay enable_debug

applicable models:

All models.

debug dhcp_relay display_dhcp_table

This command dumps the contents of the client hash table.

syntax:

display_dhcp_table

example:

Router/debug/dhcp_relay> **display_dhcp_table**

related commands:

debug dhcp_relay display_hash_statistics
debug dhcp_relay enable_debug

applicable models:

All models.

debug dhcp_relay display_hash_statistics

This command displays hash statistics for the dhcp table.

syntax:

display_hash_statistics

example:

Router/debug/dhcp_relay> **display_hash_statistics**

related commands:

debug dhcp_relay display_dhcp_table
debug dhcp_relay display enable_debug

applicable models:

All models.

debug dhcp_relay enable_debug

This command enables printing of client/relay debug messages.

syntax:

enable_debug

example:

Router/debug/dhcp_relay> **enable_debug**

related commands:

debug dhcp_relay display_dhcp_table
debug dhcp_relay display_hash_statistics

applicable models:

All models.

debug firewall

This command enables or disables firewall debugging.

parameter	definition
mode	
ad	AD debugging
algs	ALGS debugging
ipreasm	IPREASM debugging
fwatk	FWATK debugging
iapstats	IAPSTATS debugging
all	All firewall debugging

syntax:

[no] firewall mode < ad | algs | ipreasm | fwatk | iapstats | all >

example:

Router1/debug> **firewall fwatk**

applicable models:

All models.

debug fr

This command accesses next-level frame relay debug commands.

syntax:

fr

example:

Router/debug> **fr**

next-level commands

debug fr bundle-info
debug fr pvc-info
debug fr packet

applicable models:

All models.

debug fr bundle-info

This command shows frame relay interface debug information.

syntax:

debug fr bundle-info

example:

Router/debug/fr> **debug fr bundle-info**

related commands:

debug fr pvc-info
debug fr packet

applicable models:

All models.

debug fr displayifinfo

This command shows frame relay interface debug information.

syntax:

displayifinfo

example:

host/debug/fr> **displayifinfo**

related commands:

debug fr displayvcinfo
debug fr packet

applicable models:

All models.

debug fr displayvcinfo

This command shows frame relay virtual circuit debug information.

syntax:

displayvcinfo

example:

host/debug/fr> **displayvcinfo**

related commands:

debug fr displayifinfo
debug fr packet

applicable models:

All models.

debug fr mfr bundle-buffers

This command accesses next-level frame relay debug commands.

syntax:

debug fr mfr bundle-buffers

example:

Router/debug> **debug fr mfr bundle-buffers**

next-level commands

debug fr bundle-info
debug fr pvc-info
debug fr packet

applicable models:

All models.

debug fr mfr state-machine

This command accesses next-level frame relay debug commands.

syntax:

debug fr mfr state-machine

example:

Router/debug> debug fr mfr state-machine

next-level commands

debug fr bundle-info
debug fr pvc-info
debug fr packet

applicable models:

All models.

debug fr mfr states

This command accesses next-level frame relay debug commands.

parameter	definition
bundle-name	Specifies the name of the bundle.

syntax:

debug fr mfr state <bundle-name>

example:

Router/debug> **debug fr mfr states**

next-level commands

debug fr bundle-info
debug fr pvc-info
debug fr packet

applicable models:

All models.

debug fr packet

This command accesses next-level commands to enable or disable packet-level debugging.

syntax:

packet

example:

host/debug/fr> **packet**

related commands:

debug fr packet lmi

applicable models:

All models.

debug fr packet inverse-arp

This command accesses next-level commands to enable or disable packet-level debugging.

syntax:

debug fr packet inverse-arp

example:

Router/debug/fr> **debug fr packet inverse-arp**

next-level commands

debug fr packet lmi

applicable models:

All models.

debug fr packet lmi

This command shows frame relay virtual circuit information.

parameter	definition
debug_direction	
tx	Transmit direction
rx	Receive direction
both	Both transmit and receive directions This is the default.

syntax:

lmi [debug_direction]

example:

Router/debug/fr/packet> **debug fr packet lmi rx**

applicable models:

All models.

debug fr packet mfr

This command accesses next-level commands to enable or disable packet-level debugging.

syntax:

debug fr packet mfr

example:

Router/debug/fr> **debuf fr packet mfr**

next-level commands

debug fr packet lmi

applicable models:

All models.

debug fr pvc-info

This command shows frame relay virtual circuit debug information.

syntax:

debug fr pvc-info

example:

Router/debug/fr> **debug fr pvc-info**

related commands:

debug fr bundle-info
debug fr packet

applicable models:

All models.

debug framer

This command accesses next-level E1 framer commands.

syntax:

framer

example:

RouterE/debug> **framer**

next-level commands

debug framer bert
debug framer displayStatus
debug framer dumpRegister
debug framer dumpRegisters
debug framer loopInward
debug framer loopPayload
debug framer sendAllOnes
debug framer sendIdleCode
debug framer sendYellowAlarm

applicable models:

All models.

debug framer bert

This command performs a BERT test on a specified link.

Some BERT patterns may not sync to a given pattern from other test equipment. Not all equipment follows the same bit patterns specified in the standards. Patterns may also vary based on the framing mode used.

parameter	definition
link_no	Specify the link number The range is 1 - 16.
pattern	Pattern type for the bert test The default is QRW.
0s	All zeros
1s	All ones
2^11	(2^11 -1) pattern
2^15	(2^15 -1) pattern
2^23	(2^23 -1) pattern
2^20	(2^20 -1) pattern
1in7	(1 in 7) pattern
1in3	(1 in 3) pattern
QRSS	QRSS pattern (default)

syntax:

bert link_no < n > [pattern < 0s | 1s | 2^15 | 2^23 | 2^20 | 1in7 | 1in3 | QRW >]

example:

RouterE/debug/framer> **bert 4 2^23**

related commands:

debug framer displayStatus
debug framer dumpRegister
debug framer dumpRegisters
debug framer loopInward
debug framer loopPayload
debug framer sendAllOnes
debug framer sendIdleCode
debug framer sendYellowAlarm

applicable models:

All models.

debug framer displayStatus

This command shows the framer status.

syntax:

displayStatus

example:

RouterE/debug/framer> **displayStatus**

related commands:

debug framer bert
debug framer dumpRegister
debug framer dumpRegisters
debug framer loopInward
debug framer loopPayload
debug framer sendAllOnes
debug framer sendIdleCode
debug framer sendYellowAlarm

applicable models:

All models.

debug framer dumpRegister

This command shows the contents of the specified framer register.

parameter	definition
link_no	Specifies the link number The range is 1 - 16.
register_no	Specifies a register number in hex number format The range is 00-7f.

syntax:

dumpRegister <link_no < n > register_no < n >

example:

RouterE/debug/framer> **dumpRegister 5 0x01**

related commands:

debug framer bert
debug framer displayStatus
debug framer dumpRegister
debug framer dumpRegisters
debug framer loopInward
debug framer loopPayload
debug framer sendAllOnes
debug framer sendIdleCode
debug framer sendYellowAlarm

applicable models:

All models.

debug framer dumpRegisters

This command shows the content of all framer registers.

parameter	definition
link_no	Specify the link registers to dump The range is 1 - 16.

syntax:

dumpRegisters link_no < n >

example:

RouterE/debug/framer> **dumpRegisters 2**

related commands:

debug framer bert
debug framer displayStatus
debug framer dumpRegister
debug framer loopInward
debug framer loopPayload
debug framer sendAllOnes
debug framer sendIdleCode
debug framer sendYellowAlarm

applicable models:

All models.

debug framer loopInward

This command configures the framer for inward loopback.

parameter	definition
link_no	Specifies the link number The range is 1 - 16.

syntax:

loopInward link_no < n >

example:

RouterE/debug/framer> **loopInward 8**

related commands:

debug framer bert
debug framer displayStatus
debug framer dumpRegister
debug framer dumpRegisters
debug framer loopPayload
debug framer sendAllOnes
debug framer sendIdleCode
debug framer sendYellowAlarm

applicable models:

All models.

debug framer loopPayload

This command configures the framer for payload loopback.

parameter	definition
link_no	Specifies the link number The range is 1 - 16.

syntax:

loopPayload link_no < n >

example:

RouterE/debug/framer> **loopPayload 2**

related commands:

debug framer bert
debug framer displayStatus
debug framer dumpRegister
debug framer dumpRegisters
debug framer loopInward
debug framer sendAllOnes
debug framer sendIdleCode
debug framer sendYellowAlarm

applicable models:

All models.

debug framer sendAllOnes

This command sends al ones (Blue alarm) on a specified link.

parameter	definition
link_no	Specifies the link number The range is 1 - 16.

syntax:

sendAllOnes link_no < n >

example:

RouterE/debug/framer> **sendAllOnes 2**

related commands:

debug framer bert
debug framer displayStatus
debug framer dumpRegister
debug framer dumpRegisters
debug framer loopInward
debug framer loopPayload
debug framer sendIdleCode
debug framer sendYellowAlarm

applicable models:

All models.

debug framer sendIdleCode

This command sends the idle code for a specified link.

parameter	definition
link_no	Specifies the link number The range is 1 - 16.
idlecode	Specifies the idle code to be sent in hex number format

syntax:

sendIdleCode link_no < n > idlecode < n >

example:

RouterE/debug/framer> **sendIdleCode 2 0x01**

related commands:

debug framer bert
debug framer displayStatus
debug framer dumpRegister
debug framer dumpRegisters
debug framer loopInward
debug framer loopPayload
debug framer sendAllOnes
debug framer sendYellowAlarm

applicable models:

All models.

debug framer sendYellowAlarm

This command sends a yellow alarm on the specified link.

parameter	definition
link_no	Specifies the link number The range is 1 - 16.

syntax:

sendYellowAlarm link_no < n >

example:

RouterE/debug/framer> **sendYellowAlarm 10**

related commands:

debug framer bert
debug framer displayStatus
debug framer dumpRegister
debug framer dumpRegisters
debug framer loopInward
debug framer loopPayload
debug framer sendAllOnes
debug framer sendIdleCode

applicable models:

All models.

debug ip

This command accesses next-level ip/ipmux debug commands.

syntax:

ip

example:

Router/debug> **ip**

next-level commands

debug ip arp
debug ip bgp
debug ip dhcps
debug ip ospf
debug ip rip
debug ip statistics
debug ip vrrp

applicable models:

All models.

debug ip arp

This command turns arp packet debug on.

syntax:

arp

example:

Router/debug/ip> **arp**

applicable models:

All models.

debug ip bgp

This command accesses next-level BGP4 debug commands.

syntax:

bgp

example:

Router-OmniAccess 604/debug/ip> **bgp**

next-level commands

debug ip all
debug ip events
debug ip neighbor
debug ip packet
debug ip policy
debug ip routes
debug ip state
debug ip tasks
debug ip timers

applicable models:

All models.

debug ip bgp all

This command enables or disables all BGP4 debug commands.

syntax:

all

example:

Router-OmniAccess 604/debug/ip/bgp> **all**

applicable models:

All models.

debug ip bgp events

This command enables debugging of BGP4 events.

syntax:

events

example:

Router/debug/ip/bgp> **events**

applicable models:

All models.

debug ip bgp neighbor

This command displays debug information about a specified BGP4 neighbor address.

parameter	definition
address	The neighbor address for which information will be displayed.
packets	BGP packet types
open	BGP open packets
update	BGP update packets
keepalive	BGP keepalive packets
direction	Packet direction
inbound	Inbound packets
outbound	Outbound packets

syntax:

neighbor address [packets < open | update | keepalive >] [direction < inbound | outbound >]

example:

Router/debug/ip/bgp> **neighbor 132.46.23.10 open inbound**

applicable models:

All models.

debug ip bgp packet

This command accesses next-level BGP4 packet debug commands.

syntax:

packet

example:

Router-OmniAccess 604/debug/ip/bgp> **packet**

next-level commands

debug ip bgp packet all
debug ip bgp packet keepalive
debug ip bgp packet open
debug ip bgp packet update

applicable models:

All models.

debug ip bgp packet all

This command displays debug information about all BGP4 packets.

parameter	definition
direction	Packet direction
inbound	Inbound packets
outbound	Outbound packets

syntax:

all [direction < inbound | outbound >]

example:

Router-OmniAccess 604/debug/ip/bgp/packets> **all**

related commands:

debug ip bgp packet keepalive
debug ip bgp packet open
debug ip bgp packet update

applicable models:

All models.

debug ip bgp packet keepalive

This command displays debug information about BGP4 keepalive packets.

parameter	definition
direction	Packet direction
inbound	Inbound packets
outbound	Outbound packets

syntax:

keepalive [direction < inbound | outbound >]

example:

Router/debug/ip/bgp/packet> **keepalive outbound**

related commands:

debug ip bgp packet all
debug ip bgp packet open
debug ip bgp packet update

applicable models:

All models.

debug ip bgp packet open

This command displays debug information about BGP4 open packets.

parameter	definition
direction	Packet direction
inbound	Inbound packets
outbound	Outbound packets

syntax:

open [direction < inbound | outbound >]

example:

Router/debug/ip/bgp/packet> **open inbound**

related commands:

debug ip bgp packet all
debug ip bgp packet keepalive
debug ip bgp packet update

applicable models:

All models.

debug ip bgp packet update

This command displays debug information about BGP4 update packets.

parameter	definition
direction	Packet direction
inbound	Inbound packets
outbound	Outbound packets

syntax:

update [direction < inbound | outbound >]

example:

Router/debug/ip/bgp/packet> **update**

related commands:

debug ip bgp packet all
debug ip bgp packet keepalive
debug ip bgp packet open

applicable models:

All models.

debug ip bgp policy

This command displays debug information about BGP4 policy configuration.

syntax:

policy

example:

Router/debug/ip/bgp> **policy**

applicable models:

All models.

debug ip bgp routes

This command displays debug information about BGP4 routes.

syntax:

routes

example:

Router/debug/ip/bgp> **routes**

applicable models:

All models.

debug ip bgp state

This command displays debug information about BGP4 state machine transitions.

syntax:

state

example:

Router/debug/ip/bgp> **state**

applicable models:

All models.

debug ip bgp tasks

This command displays debug information about BGP4 tasks.

syntax:

tasks

example:

Router/debug/ip/bgp> **tasks**

applicable models:

All models.

debug ip bgp timers

This command displays debug information about BGP4 timers.

syntax:

timers

example:

Router/debug/ip/bgp> **timers**

applicable models:

All models.

debug ip dhcps

This command accesses next-level commands to enable or disable DHCP server debug commands.

syntax:

dhcps

example:

Router/debug/ip> **dhcps**

next-level commands

debug ip dhcps all
debug ip dhcps error
debug ip dhcps events
debug ip dhcps packet
debug ip dhcps state

applicable models:

All models.

debug ip dhcps all

This command enables or disables all DHCP server debug messages.

syntax:

[no] all

example:

Router/debug/ip/dhcps> **all**

related commands:

debug ip dhcps error
debug ip dhcps events
debug ip dhcps packet
debug ip dhcps state

applicable models:

All models.

debug ip dhcps error

This command enables or disables DHCP server error debug messages.

syntax:

error

example:

Router/debug/ip/dhcps> **error**

related commands:

debug ip dhcps all
debug ip dhcps events
debug ip dhcps packet
debug ip dhcps state

applicable models:

All models.

debug ip dhcps events

This command enables or disables debug information about all DHCP events.

syntax:

events

example:

Router/debug/ip/dhcps> **events**

related commands:

debug ip dhcps all
debug ip dhcps error
debug ip dhcps packet
debug ip dhcps state

applicable models:

All models.

debug ip dhcps packet

This command enables or disables debug information about all DHCP packet messages.

syntax:

packet

example:

Router/debug/ip/dhcps> **packet**

related commands:

debug ip dhcps all
debug ip dhcps error
debug ip dhcps events
debug ip dhcps state

applicable models:

All models.

debug ip dhcps state

This command enables or disables debug information about all DHCP state transitions.

syntax:

state

example:

Router/debug/ip/dhcps> **state**

related commands:

debug ip dhcps all
debug ip dhcps error
debug ip dhcps events
debug ip dhcps packet

applicable models:

All models.

debug ip ospf

This command accesses next-level commands to enable or disable OSPF debug commands.

NOTE: The router must be in routing mode to access this debug command.

syntax:

ospf

example:

Router/debug/ip> **ospf**

next-level commands

debug ip ospf all
debug ip ospf database
debug ip ospf dr_election
debug ip ospf flooding
debug ip ospf packet
debug ip ospf policy
debug ip ospf spf
debug ip ospf spf_timing
debug ip ospf state_changes
debug ip ospf summary

applicable models:

All models.

debug ip ospf all

This command enables or disables debug messages for all OSPF events.

NOTE: The router must be in routing mode to access this debug command.

syntax:

all

example:

Router/debug/ip> **ospf all**

related commands:

debug ip ospf database
debug ip ospf dr_election
debug ip ospf flooding
debug ip ospf packet
debug ip ospf policy
debug ip ospf spf
debug ip ospf spf_timing
debug ip ospf state_changes
debug ip ospf summary

applicable models:

All models.

debug ip ospf database

This command enables or disables debug messages relating to the OSPF database operation.

NOTE: The router must be in routing mode to access this debug command.

syntax:

database

example:

Router/debug/ip> **ospf database**

related commands:

debug ip ospf all
debug ip ospf dr_election
debug ip ospf flooding
debug ip ospf packet
debug ip ospf policy
debug ip ospf spf
debug ip ospf spf_timing
debug ip ospf state_changes
debug ip ospf summary

applicable models:

All models.

debug ip ospf dr_election

This command enables or disables debug messages for the OSPF designated router election.

NOTE: The router must be in routing mode to access this debug command.

syntax:

dr_election

example:

Router/debug/ip> **ospf dr_election**

related commands:

debug ip ospf all
debug ip ospf database
debug ip ospf flooding
debug ip ospf packet
debug ip ospf policy
debug ip ospf spf
debug ip ospf spf_timing
debug ip ospf state_changes
debug ip ospf summary

applicable models:

All models.

debug ip ospf flooding

This command enables or disables debug messages for OSPF flooding.

NOTE: The router must be in routing mode to access this debug command.

syntax:

flooding

example:

Router/debug/ip> **ospf flooding**

related commands:

debug ip ospf all
debug ip ospf database
debug ip ospf dr_election
debug ip ospf packet
debug ip ospf policy
debug ip ospf spf
debug ip ospf spf_timing
debug ip ospf state_changes
debug ip ospf summary

applicable models:

All models.

debug ip ospf packet

This command accesses next-level commands to enable or disable debug messages for OSPF packets.

NOTE: The router must be in routing mode to access this debug command.

syntax:

packet

example:

Router/debug/ip> **ospf packet**

next-level commands

debug ip ospf packet all
debug ip ospf packet dd
debug ip ospf packet hello
debug ip ospf packet ls_ack
debug ip ospf packet ls_request
debug ip ospf packet ls_update

applicable models:

All models.

debug ip ospf packet all

This command accesses next-level commands to enable or disable debug messages for all OSPF packets.

NOTE: The router must be in routing mode to access this debug command.

parameter	definition
direction	Packet direction
inbound	Inbound packets
outbound	Outbound packets

syntax:

all [direction < inbound | outbound >]

example:

Router/debug/ip> **ospf packet all outbound**

related commands:

debug ip ospf packet dd
debug ip ospf packet hello
debug ip ospf packet ls_ack
debug ip ospf packet ls_request
debug ip ospf packet ls_update

applicable models:

All models.

debug ip ospf packet dd

This command enables or disables debug messages for OSPF database description packets.

NOTE: The router must be in routing mode to access this debug command.

parameter	definition
direction	Packet direction
inbound	Inbound packets
outbound	Outbound packets

syntax:

dd [direction < inbound | outbound >]

example:

Router/debug/ip> **ospf packet dd**

related commands:

debug ip ospf packet all
debug ip ospf packet hello
debug ip ospf packet ls_ack
debug ip ospf packet ls_request
debug ip ospf packet ls_update

applicable models:

All models.

debug ip ospf packet hello

This command enables or disables debug messages for OSPF hello packets.

NOTE: The router must be in routing mode to access this debug command.

parameter	definition
direction	Packet direction
inbound	Inbound packets
outbound	Outbound packets

syntax:

hello [direction < inbound | outbound >]

example:

Router/debug/ip> **ospf packet hello outbound**

related commands:

debug ip ospf packet all
debug ip ospf packet dd
debug ip ospf packet ls_ack
debug ip ospf packet ls_request
debug ip ospf packet ls_update

applicable models:

All models.

debug ip ospf packet ls_ack

This command enables or disables debug messages for OSPF link state acknowledgement packets.

NOTE: The router must be in routing mode to access this debug command.

parameter	definition
direction	Packet direction
inbound	Inbound packets
outbound	Outbound packets

syntax:

ls_ack [direction < inbound | outbound >]

example:

Router/debug/ip> **ospf packet ls_ack**

related commands:

debug ip ospf packet all
debug ip ospf packet dd
debug ip ospf packet hello
debug ip ospf packet ls_request
debug ip ospf packet ls_update

applicable models:

All models.

debug ip ospf packet ls_request

This command enables or disables debug messages for OSPF link state request packets.

NOTE: The router must be in routing mode to access this debug command.

parameter	definition
direction	Packet direction
inbound	Inbound packets
outbound	Outbound packets

syntax:

ls_request [direction < inbound | outbound >]

example:

Router/debug/ip> **ospf packet ls_request**

related commands:

debug ip ospf packet all
debug ip ospf packet dd
debug ip ospf packet hello
debug ip ospf packet ls_ack
debug ip ospf packet ls_update

applicable models:

All models.

debug ip ospf packet ls_update

This command enables or disables debug messages for OSPF link state update packets.

NOTE: The router must be in routing mode to access this debug command.

parameter	definition
direction	Packet direction
inbound	Inbound packets
outbound	Outbound packets

syntax:

ls_update [direction < inbound | outbound >]

example:

Router/debug/ip> **ospf packet ls_update**

related commands:

debug ip ospf packet all
debug ip ospf packet dd
debug ip ospf packet hello
debug ip ospf packet ls_ack
debug ip ospf packet ls_request

applicable models:

All models.

debug ip ospf policy

This command enables or disables debug messages for OSPF policy.

NOTE: The router must be in routing mode to access this debug command.

syntax:

policy

example:

Router/debug/ip> **ospf policy**

related commands:

debug ip ospf all
debug ip ospf database
debug ip ospf dr_election
debug ip ospf flooding
debug ip ospf packet
debug ip ospf spf
debug ip ospf spf_timing
debug ip ospf state_changes
debug ip ospf summary

applicable models:

All models.

debug ip ospf spf

This command enables or disables debug messages for OSPF spf.

NOTE: The router must be in routing mode to access this debug command.

syntax:

spf

example:

Router/debug/ip> **ospf spf**

related commands:

debug ip ospf all
debug ip ospf database
debug ip ospf dr_election
debug ip ospf flooding
debug ip ospf packet
debug ip ospf policy
debug ip ospf spf_timing
debug ip ospf state_changes
debug ip ospf summary

applicable models:

All models.

debug ip ospf spf_timing

This command enables or disables SPF timing measurement.

NOTE: The router must be in routing mode to access this debug command.

syntax:

spf_timing

example:

Router/debug/ip> **ospf spf_timing**

related commands:

debug ip ospf all
debug ip ospf database
debug ip ospf dr_election
debug ip ospf flooding
debug ip ospf packet
debug ip ospf policy
debug ip ospf spf
debug ip ospf state_changes
debug ip ospf summary

applicable models:

All models.

debug ip ospf state_changes

This command enables or disables debug messages for OSPF state changes.

NOTE: The router must be in routing mode to access this debug command.

syntax:

state_changes

example:

Router/debug/ip> **ospf state_changes**

related commands:

debug ip ospf all
debug ip ospf database
debug ip ospf dr_election
debug ip ospf flooding
debug ip ospf packet
debug ip ospf policy
debug ip ospf spf
debug ip ospf spf_timing
debug ip ospf summary

applicable models:

All models.

debug ip ospf summary

This command enables or disables debug messages for OSPF summarization

NOTE: The router must be in routing mode to access this debug command.

syntax:

summary

example:

Router/debug/ip> **ospf summary**

related commands:

debug ip ospf all
debug ip ospf database
debug ip ospf dr_election
debug ip ospf flooding
debug ip ospf packet
debug ip ospf policy
debug ip ospf spf
debug ip ospf spf_timing
debug ip ospf state_changes

applicable models:

All models.

debug ip rip

This command accesses next-level commands for enabling or disabling RIP debug commands.

NOTE: The router must be in routing mode to access this debug command.

syntax:

rip

example:

Router/debug/ip> **rip**

next-level commands

debug ip rip all
debug ip rip detail
debug ip rip flood
debug ip rip packet
debug ip rip state

applicable models:

All models.

debug ip rip all

This command enables or disables all RIP debug commands.

NOTE: The router must be in routing mode to access this debug command.

syntax:

[no] all

example:

Router/debug/ip> **rip all**

related commands:

debug ip rip detail
debug ip rip flood
debug ip rip packet
debug ip rip state

applicable models:

All models.

debug ip rip detail

This command enables or disables RIP debug detail messages.

NOTE: The router must be in routing mode to access this debug command.

syntax:

detail

example:

Router/debug/ip> **rip detail**

related commands:

debug ip rip all
debug ip rip flood
debug ip rip packet
debug ip rip state

applicable models:

All models.

debug ip rip flood

This command enables or disables all RIP debug flooding messages.

NOTE: The router must be in routing mode to access this debug command.

syntax:

flood

example:

Router/debug/ip> **rip flood**

related commands:

debug ip rip all
debug ip rip detail
debug ip rip packet
debug ip rip state

applicable models:

All models.

debug ip rip packet

This command enables or disables all RIP debug packet messages.

NOTE: The router must be in routing mode to access this debug command.

parameter	definition
pkt_type	The RIP debug packet type
all	All packets
send	Packets sent
receive	Packets received
summary	Summary of updates
interface_name	The name of the RIP interface
dlci	The data link connection identifier of the PVC The range is 16 - 1022.

syntax:

packet pkt_type < all | send | receive | summary > [interface_name < name >] [dlci < n >]

example:

Router/debug/ip> **rip packet all**

related commands:

debug ip rip all
debug ip rip detail
debug ip rip flood
debug ip rip state

applicable models:

All models.

debug ip rip state

This command enables or disables all RIP debug state messages.

NOTE: The router must be in routing mode to access this debug command.

syntax:

state

example:

Router/debug/ip> **rip state**

related commands:

debug ip rip all
debug ip rip detail
debug ip rip flood
debug ip rip packet

applicable models:

All models.

debug ip statistics

This command accesses next-level commands for showing various kinds of debug statistics.

syntax:

statistics

example:

Router/debug/ip> **statistics**

next-level commands

debug ip statistics icmpshow
debug ip statistics ipmuxclear
debug ip statistics ipmuxshow
debug ip statistics ipshow
debug ip statistics rtshow
debug ip statistics tcpshow
debug ip statistics udpshow

applicable models:

All models.

debug ip statistics icmpshow

This command displays icmp statistics.

syntax:

icmpshow

example:

Router/debug/ip> **statistics icmpshow**

related commands:

debug ip statistics ipmuxclear
debug ip statistics ipmuxshow
debug ip statistics ipshow
debug ip statistics rtshow
debug ip statistics tcpshow
debug ip statistics udpshow

applicable models:

All models.

debug ip statistics ipmuxclear

This command clears ipmux debug statistics.

syntax:

ipmuxclear

example:

Router/debug/ip> **statistics ipmuxclear**

related commands:

debug ip statistics icmpshow
debug ip statistics ipmuxshow
debug ip statistics ipshow
debug ip statistics rtshow
debug ip statistics tcpshow
debug ip statistics udpshow

applicable models:

All models.

debug ip statistics ipmuxshow

This command shows ipmux debug statistics.

syntax:

ipmuxshow

example:

Router/debug/ip> **statistics ipmuxshow**

related commands:

debug ip statistics icmpshow
debug ip statistics ipmuxclear
debug ip statistics ipshow
debug ip statistics rtshow
debug ip statistics tcpshow
debug ip statistics udpshow

applicable models:

All models.

debug ip statistics ipshow

This command shows ip debug statistics.

syntax:

statistics ipshow

example:

Router/debug/ip> **statistics ipshow**

related commands:

debug ip statistics icmpshow
debug ip statistics ipmuxclear
debug ip statistics ipmuxshow
debug ip statistics rtshow
debug ip statistics tcpshow
debug ip statistics udpshow

applicable models:

All models.

debug ip statistics rtshow

This command shows debug route statistics.

syntax:

rtshow

example:

Router/debug/ip> **statistics rtshow**

related commands:

debug ip statistics icmpshow
debug ip statistics ipmuxclear
debug ip statistics ipmuxshow
debug ip statistics ipshow
debug ip statistics tcpshow
debug ip statistics udpshow

applicable models:

All models.

debug ip tcpshow

This command shows tcp debug statistics.

syntax:

tcpshow

example:

Router/debug/ip> **statistics tcpshow**

related commands:

debug ip statistics icmpshow
debug ip statistics ipmuxclear
debug ip statistics ipmuxshow
debug ip statistics ipshow
debug ip statistics rtshow
debug ip statistics udpshow

applicable models:

All models.

debug ip udpshow

This command shows udp debug statistics.

syntax:

udpshow

example:

Router/debug/ip> **statistics udpshow**

related commands:

debug ip statistics icmpshow
debug ip statistics ipmuxclear
debug ip statistics ipmuxshow
debug ip statistics ipshow
debug ip statistics rtshow
debug ip statistics tcpshow

applicable models:

All models.

debug ip vrrp

This command accesses next-level commands for enabling or disabling VRRP debug commands.

NOTE: The router must be in routing mode to access this debug command.

syntax:

vrrp

example:

Router/debug/ip> **vrrp**

next-level commands

debug ip vrrp all
debug ip vrrp error
debug ip vrrp events
debug ip vrrp packet
debug ip vrrp state

applicable models:

All models.

debug ip vrrp all

This command enables or disables all VRRP debug messages.

NOTE: The router must be in routing mode to access this debug command.

syntax:

[no] all

example:

Router/debug/ip> **vrrp all**

related commands:

debug ip vrrp error
debug ip vrrp events
debug ip vrrp packet
debug ip vrrp state

applicable models:

All models.

debug ip vrrp error

This command enables or disables all VRRP error debug messages.

NOTE: The router must be in routing mode to access this debug command.

syntax:

error

example:

Router/debug/ip> **vrrp error**

related commands:

debug ip vrrp all
debug ip vrrp events
debug ip vrrp packet
debug ip vrrp state

applicable models:

All models.

debug ip vrrp events

This command enables or disables all VRRP events debug messages.

NOTE: The router must be in routing mode to access this debug command.

syntax:

events

example:

Router/debug/ip> **vrrp events**

related commands:

debug ip vrrp all
debug ip vrrp error
debug ip vrrp packet
debug ip vrrp state

applicable models:

All models.

debug ip vrrp packet

This command enables or disables all VRRP packet debug messages.

NOTE: The router must be in routing mode to access this debug command.

parameter	definition
pkt_type	The VRRP packet type
all	All packets
arp	ARP packets
send	Sent packets
recv	Received packets

syntax:

packet < pkt_type all | arp | send | recv >

example:

Router/debug/ip> **vrrp packet arp**

related commands:

debug ip vrrp all
debug ip vrrp error
debug ip vrrp events
debug ip vrrp state

applicable models:

All models.

debug ip vrrp state

This command enables or disables VRRP state transition debug commands.

NOTE: The router must be in routing mode to access this debug command.

syntax:

state

example:

Router/debug/ip> **vrrp state**

related commands:

debug ip vrrp all
debug ip vrrp error
debug ip vrrp events
debug ip vrrp packet

applicable models:

All models.

debug lance

This command accesses next-level Ethernet debug commands.

syntax:

lance

example:

Router/debug> **lance**

next-level commands

debug lance errclear
debug lance errshow
debug lance statshow

applicable models:

All models.

debug lance errclear

This command clears lance errors.

syntax:

errclear

example:

Router/debug/lance> **errclear**

related commands:

debug lance errshow
debug lance statshow

applicable models:

All models.

debug lance errshow

This command displays lance errors.

syntax:

errshow

example:

Router/debug/lance> **errshow**

related commands:

debug lance errclear
debug lance statshow

applicable models:

All models.

debug lance statshow

This command displays lance statistics.

parameter	definition
ethernet_number	The number of the Ethernet interface The range is 0 - 1; the default is 0.

syntax:

statshow [ethernet_number < 0 | 1 >]

example:

Router/debug/lance> **statshow**

related commands:

debug lance errclear
debug lance errshow

applicable models:

All models.

debug mhu

This command accesses next-level commands to enable or disable MHU debug commands.

syntax:

mhu

example:

Router/debug> **mhu**

next-level commands

debug mhu all
debug mhu cleanup
debug mhu config
debug mhu dhcp
debug mhu disp_redirect_entries
debug mhu filters
debug mhu hash
debug mhu redirect
debug mhu snmp
debug mhu static _host

applicable models:

All models.

debug mhu all

This command enables or disables all MHU debug messages.

syntax:

all

example:

Router/debug/mhu> **all**

related commands:

debug mhu cleanup
debug mhu config
debug mhu dhcp
debug mhu disp_redirect_entries
debug mhu filters
debug mhu hash
debug mhu redirect
debug mhu snmp
debug mhu static _host

applicable models:

All models.

debug mhu cleanup

This command enables or disables MHU debug messages for cleanup.

syntax:

cleanup

example:

Router/debug/mhu> **cleanup**

related commands:

debug mhu all
debug mhu config
debug mhu dhcp
debug mhu disp_redirect_entries
debug mhu filters
debug mhu hash
debug mhu redirect
debug mhu snmp
debug mhu static _host

applicable models:

All models.

debug mhu config

This command enables or disables MHU debug messages for configure commands.

syntax:

config

example:

Router/debug/mhu> **config**

related commands:

debug mhu all
debug mhu cleanup
debug mhu dhcp
debug mhu disp_redirect_entries
debug mhu filters
debug mhu hash
debug mhu redirect
debug mhu snmp
debug mhu static _host

applicable models:

All models.

debug mhu dhcp

This command enables or disables MHU debug messages for dhcp.

syntax:

dhcp

example:

Router/debug/mhu> **dhcp**

related commands:

debug mhu all
debug mhu cleanup
debug mhu config
debug mhu disp_redirect_entries
debug mhu filters
debug mhu hash
debug mhu redirect
debug mhu snmp
debug mhu static _host

applicable models:

All models.

debug mhu disp_redirect_entries

This command displays the MHU redirect table.

syntax:

disp_redirect_entries

example:

Router/debug/mhu> **disp_redirect_entries**

related commands:

debug mhu all
debug mhu cleanup
debug mhu config
debug mhu dhcp
debug mhu filters
debug mhu hash
debug mhu redirect
debug mhu snmp
debug mhu static _host

applicable models:

All models.

debug mhu filters

This command enables or disables MHU debug messages for filters.

syntax:

filters

example:

Router/debug/mhu> **filters**

related commands:

debug mhu all
debug mhu cleanup
debug mhu config
debug mhu dhcp
debug mhu disp_redirect_entries
debug mhu hash
debug mhu redirect
debug mhu snmp
debug mhu static _host

applicable models:

All models.

debug mhu hash

This command displays the MHU hash table.

syntax:

hash

example:

Router/debug/mhu> **hash**

related commands:

debug mhu all
debug mhu cleanup
debug mhu config
debug mhu dhcp
debug mhu disp_redirect_entries
debug mhu filters
debug mhu redirect
debug mhu snmp
debug mhu static _host

applicable models:

All models.

debug mhu redirect

This command enables or disables MHU debug messages for redirection.

syntax:

redirect

example:

Router/debug/mhu> **redirect**

related commands:

debug mhu all
debug mhu cleanup
debug mhu config
debug mhu dhcp
debug mhu disp_redirect_entries
debug mhu filters
debug mhu hash
debug mhu snmp
debug mhu static _host

applicable models:

All models.

debug mhu snmp

This command enables or disables MHU debug messages for snmp.

syntax:

snmp

example:

Router/debug/mhu> **snmp**

related commands:

debug mhu all
debug mhu cleanup
debug mhu config
debug mhu dhcp
debug mhu disp_redirect_entries
debug mhu filters
debug mhu hash
debug mhu redirect
debug mhu static _host

applicable models:

All models.

debug mhu static_host

This command enables or disables MHU debug messages for static hosts.

syntax:

static_host

example:

Router/debug/mhu> **static_host**

related commands:

debug mhu all
debug mhu cleanup
debug mhu config
debug mhu dhcp
debug mhu disp_redirect_entries
debug mhu filters
debug mhu hash
debug mhu redirect
debug mhu snmp

applicable models:

All models.

debug nat

This command accesses next-level commands to enable or disable NAT debug commands.

syntax:

nat

example:

Router/debug> **nat**

next-level commands

debug nat global
debug nat interface

applicable models:

All models.

debug nat ethernet

This command enables NAT on an Ethernet interface and accesses next-level commands.

syntax:

interface ethernet < 0 | 1 >

example:

Router/debug/nat> **interface ethernet 0**

next-level commands

debug nat ethernet debug
debug nat ethernet hash
debug nat ethernet packet

applicable models:

All models.

debug nat ethernet debug

This command enables or disables internal NAT debug messages to the console.

syntax:

debug

example:

Router/debug/nat/interface etherent 0> **debug**

related commands:

debug nat ethernet hash
debug nat ethernet packet

applicable models:

All models.

debug nat ethernet hash

This command dumps the NAT Ethernet hash table to the management console.

syntax:

hash

example:

Router/debug/nat/interface etherent 0> **hash**

related commands:

debug nat ethernet debug
debug nat ethernet packet

applicable models:

All models.

debug nat ethernet packet

This command enables or disables NAT Ethernet packet debug messages to the management console.

syntax:

packet

example:

Router/debug/nat/interface etherent 0> **packet**

related commands:

debug nat ethernet debug
debug nat ethernet hash

applicable models:

All models.

debug nat global

This command accesses next-level commands to enable debug commands for global NAT.

syntax:

global

example:

Router/debug/nat> **global**

next-level commands

debug nat global debug
debug nat global hash
debug nat global packet

applicable models:

All models.

debug nat global debug

This command enables or disables internal NAT debug messages to the management console.

syntax:

debug

example:

Router/debug/nat/global> **debug**

related commands:

debug nat global hash
debug nat global packet

applicable models:

All models.

debug nat global hash

This command displays the NAT hash table to the management console.

syntax:

hash

example:

Router/debug/nat/global> **hash**

related commands:

debug nat global debug
debug nat global packet

applicable models:

All models.

debug nat global packet

This command enables or disables NAT packet debug messages to the management console.

syntax:

packet

example:

Router/debug/nat/global> **packet**

related commands:

debug nat global debug
debug nat global hash

applicable models:

All models.

debug nat interface

This command accesses next-level commands to enable NAT for a specific interface.

syntax:

interface

example:

Router/debug/nat> **interface**

next-level commands

debug nat interface bundle
debug nat interface ethernet

applicable models:

All models.

debug nat interface bundle

This command enables NAT debug on the bundle interface and accesses next-level commands.

parameter	definition
bundle_name	Name of the bundle Use a maximum of eight characters. For frame relay bundles, use: bundle_name:pvc.

syntax:

bundle < bundle_name >

example:

Router/debug/nat/interface> **bundle northwest**

next-level commands

debug nat interface bundle debug
debug nat interface bundle hash
debug nat interface bundle packet

applicable models:

All models.

debug nat interface bundle debug

This command enables or disables internal NAT debug messages to the management console.

syntax:

debug

example:

Router/debug/nat/interface/bundle northwest> **debug**

related commands:

debug nat interface bundle hash
debug nat interface bundle packet

All models.

debug nat interface bundle hash

This command dumps the NAT hash table to the management console.

syntax:

hash

example:

Router/debug/nat/interface/bundle northwest> **hash**

related commands:

debug nat interface bundle debug
debug nat interface bundle packet

applicable models:

All models.

debug nat interface bundle packet

This command enables or disables NAT packet debug messages to the management console.

syntax:

packet

example:

Router/debug/nat/interface/bundle northwest> **packet**

related commands:

debug nat interface bundle debug
debug nat interface bundle hash

applicable models:

All models.

debug ppp

This command accesses next-level commands to enable or disable PPP debug messages.

syntax:

ppp

example:

Router/debug> **ppp**

next-level commands

debug ppp bcp
debug ppp ipcp
debug ppp lcp
debug ppp mlpinfo
debug ppp negotiation
debug ppp pppstates

applicable models:

All models.

debug ppp bcp

This command shows BCP debug information.

syntax:

bcp

example:

Router/debug/ppp> **bcp**

related commands:

debug ppp ipcp
debug ppp lcp
debug ppp mlpinfo
debug ppp negotiation
debug ppp pppstates

applicable models:

All models.

debug ppp debug_link

This command provides the ability to enable link level debugging on a per-link basis when multiple PPP/MLPPP bundles are configured.

NOTE: This command will only work for LCP and only one link can be debugged at any given time.

Enabling the link level debug of a PPP/MLPPP bundle is a three-step process:

1 Find the link number of the link that you want to debug.

To find the link, issue the debug ppp pppstates command specifying the bundle name.

```
host/debug> ppp pppstates wan2

link        LCP State       Link No
----        ---------       -------
t1 3        S6(St_ReqSent)  0x00010001
t1 4        S6(St_ReqSent)  0x00010002

IPCP State: S0(St_Initial)

BCP State: S0(St_Initial)
```

2 Enable the link level debug using the link number obtained in step 1.

Enabling link level debug only sets the link to be debugged and will not enable ppp level debugging. To set the link debug, issue the debug ppp debug_link command specifying the link number obtained in step 1.

3 Enable lcp debug for PPP/MLPPP. Issue the **debug ppp lcp** command.

This step is required to see any output on the screen.

parameter	definition
link_num	Link number in hexadecimal (no 0x prefix)

syntax:

[no] debug_link link_num < n >

example:

Router/debug> **ppp debug_link 00010001**

applicable models:

All models.

debug ppp ipcp

This command shows IPCP debug information.

syntax:

ipcp

example:

Router/debug/ppp> **ipcp**

related commands:

debug ppp bcp
debug ppp lcp
debug ppp mlpinfo
debug ppp negotiation
debug ppp pppstates

applicable models:

All models.

debug ppp lcp

This command shows LCP debug information.

syntax:

lcp

example:

Router/debug/ppp> **lcp**

related commands:

debug ppp bcp
debug ppp ipcp
debug ppp mlpinfo
debug ppp negotiation
debug ppp pppstates

applicable models:

All models.

debug ppp mlpinfo

This command shows MLPPP information for a specified bundle.

parameter	definition
bundle_name	Name of the bundle for which debug information will be obtained.

syntax:

mlpinfo bundle_name < name >

example:

Router/debug/ppp> **mlpinfo northwest**

related commands:

debug ppp bcp
debug ppp ipcp
debug ppp icp
debug ppp negotiation
debug ppp pppstates

applicable models:

All models.

debug ppp negotiation

This command shows PPP the debug information associated with PPP negotiation (LCP/IPCP/BCP).

syntax:

negotiation

example:

Router/debug/ppp> **negotiation**

related commands:

debug ppp bcp
debug ppp ipcp
debug ppp icp
debug ppp mlpinfo
debug ppp pppstates

applicable models:

All models.

debug ppp pppstates

This command enables or disables PPP debug commands to show the PPP states of all links in the bundle.

parameter	definition
bundle_name	Name of the bundle for which debug pppstates information will be obtained.

syntax:

pppstates bundle_name < name >

example:

Router/debug/ppp> **pppstates northwest**

related commands:

debug ppp bcp
debug ppp ipcp
debug ppp icp
debug ppp mlpinfo
debug ppp negotiation

applicable models:

All models.

debug qos

This command accesses next-level commands for debugging QoS functionality.

syntax:

qos

example:

Router/debug> **qos**

next-level commands

debug qos buf_mgmt_info
debug qos clear_buf_overruns
debug qos clear_sch_info
debug qos clear_upload_counters
debug qos hist_stats_upload_info
debug qos show_buf_overruns
debug qos show_intf_qos_info
debug qos show_sch_info
debug qos show_sch_list

applicable models:

All models.

debug qos buf_mgmt_info

This command shows buffer management information for the specified bundle.

When the command is given without any parameters, the global buffer management information is displayed. When the bundle name is specified (and PVC number, if applicable) buffer allocation information for the bundle (or PVC) is also displayed.

parameter	definition
bundle_name	The name of the bundle.
pvc	For frame relay, the pvc number

syntax:

buf_mgmt_info [bundle_name < name >] [pvc < number >]

example 1:

Router/debug/qos> **buf_mgmt_info**

example 2:

Router/debug/qos> **buf_mgmt_info bundle wan1**

example 3:

Router/debug/qos> **buf_mgmt_info bundle fr1 pvc 99**

related commands:

debug qos clear_buf_overruns
debug qos clear_sch_info
debug qos clear_upload_counters
debug qos hist_stats_upload_info
debug qos show_buf_overruns
debug qos show_intf_qos_info
debug qos show_sch_info
debug qos show_sch_list

applicable models:

All models.

debug qos clear_buf_overruns

This command clears common stack buffer overrun counters.

syntax:

clear_buf_overruns

example:

Router/debug/qos> **clear_buf_overrun**s

related commands:

debug qos buf_mgmt_info
debug qos clear_sch_info
debug qos clear_upload_counters
debug qos hist_stats_upload_info
debug qos show_buf_overruns
debug qos show_intf_qos_info
debug qos show_sch_info
debug qos show_sch_list

applicable models:

All models.

debug qos clear_sch_info

This command clears and resets scheduler debug information.

syntax:

clear_sch_info

example:

Router/debug/qos> **clear_sch_info**

related commands:

debug qos buf_mgmt_info
debug qos clear_buf_overruns
debug qos clear_upload_counters
debug qos hist_stats_upload_info
debug qos show_buf_overruns
debug qos show_intf_qos_info
debug qos show_sch_info
debug qos show_sch_list

applicable models:

All models.

debug qos clear_upload_counters

This command clears debug statistics related to uploading of historical statistics.

syntax:

clear_upload_counters

example:

Router/debug/qos> **clear_upload_counters**

related commands:

debug qos buf_mgmt_info
debug qos clear_buf_overruns
debug qos clear_sch_info
debug qos hist_stats_upload_info
debug qos show_buf_overruns
debug qos show_intf_qos_info
debug qos show_sch_info
debug qos show_sch_list

applicable models:

All models.

debug qos hist_stats_upload_info

This command shows debug information related to uploading of historical statistics.

syntax:

hist_stats_upload_info

example:

Router/debug/qos> **hist_stats_upload_info**

related commands:

debug qos buf_mgmt_info
debug qos clear_buf_overruns
debug qos clear_sch_info
debug qos clear_upload_counters
debug qos show_buf_overruns
debug qos show_intf_qos_info
debug qos show_sch_info
debug qos show_sch_list

applicable models:

All models.

debug qos show_buf_overruns

This command shows receive buffer overruns of the LAN and WAN due to insufficient buffers on the common stack.

syntax:

show_buf_overruns

example:

Router/debug/qos> **show_buf_overruns**

related commands:

debug qos buf_mgmt_info
debug qos clear_buf_overruns
debug qos clear_sch_info
debug qos clear_upload_counters
debug qos hist_stats_upload_info
debug qos show_intf_qos_info
debug qos show_sch_info
debug qos show_sch_list

applicable models:

All models.

debug qos show_intf_qos_info

This command shows debug QoS configuration information for the interface.

It displays "root class" information, the interface class list, and other details.

syntax:

show_intf_qos_info

example:

Router/debug/qos> **show_intf_qos_info**

related commands:

debug qos buf_mgmt_info
debug qos clear_buf_overruns
debug qos clear_sch_info
debug qos clear_upload_counters
debug qos hist_stats_upload_info
debug qos show_buf_overruns
debug qos show_sch_info
debug qos show_sch_list

applicable models:

All models.

debug qos show_sch_info

This command shows scheduler debug information.

It shows the number of times that the scheduler could not service the queues in time (the maximum recorded time for one scheduler pass etc.).

syntax:

show_sch_info

example:

Router/debug/qos> **show_sch_info**

related commands:

debug qos buf_mgmt_info
debug qos clear_buf_overruns
debug qos clear_sch_info
debug qos clear_upload_counters
debug qos hist_stats_upload_info
debug qos show_buf_overruns
debug qos show_intf_qos_info
debug qos show_sch_list

applicable models:

All models.

debug qos show_sch_list

This command shows the current snapshot of the scheduler list on the specified interface.

It shows how the classes on an interface are placed in the scheduler list.

syntax:

show_sch_list < bundle_name >

example:

Router/debug/qos> **show_sch_list wan1**

related commands:

debug qos buf_mgmt_info
debug qos clear_buf_overruns
debug qos clear_sch_info
debug qos clear_upload_counters
debug qos hist_stats_upload_info
debug qos show_buf_overruns
debug qos show_intf_qos_info
debug qos show_sch_info

applicable models:

All models.

debug rtp

This command accesses next-level commands display RTP debug information.

syntax:

rtp

example:

Router/debug> **rtp**

next-level commands

debug rtp rxtable
debug rtp statistics
debug rtp tables
debug rtp txtable

applicable models:

All models.

debug rtp rxtable

This command displays the RTP receive table for the specified bundle interface.

syntax:

rxtable interface < name >

example:

Router/debug/rtp> **rxtable northwest**

related commands:

debug rtp statistics
debug rtp tables
debug rtp txtable

applicable models:

All models.

debug rtp statistics

This command displays RTP statistics for the specified bundle interface.

syntax:

statistics interface < name >

example:

Router/debug/rtp> **statistics northwest**

related commands:

debug rtp rxtable
debug rtp tables
debug rtp txtable

applicable models:

All models.

debug rtp tables

This command displays the transmit and receive tables for the specified bundle interface.

syntax:

tables interface < name >

example:

Router/debug/rtp> **tables northwest**

related commands:

debug rtp rxtable
debug rtp statistics
debug rtp txtable

applicable models:

All models.

debug rtp txtable

This command displays the transmit table for the specified bundle interface.

syntax:

txtable interface < name >

example:

Router/debug/rtp> **txtable northwest**

related commands:

debug rtp rxtable
debug rtp statistics
debug rtp tables

applicable models:

All models.

debug system

This command accesses next-level commands to show system debug information.

syntax:

system

example:

Router/debug> **system**

next-level commands

debug system clear_crash_dump
debug system clear_stats
debug system datapool_display
debug system display_loaned_common_buffer
debug system display_overwrite_crash_dump
debug system overwrite_crash_dump
debug system print_stats
debug system show_crash
debug system stackpool_display

applicable models:

All models.

debug system clear_crash_dump

This command clears all crash dump information.

syntax:

clear_crash_dump

example:

Router/debug/system> **clear_crash_dump**

related commands:

debug system clear_stats
debug system datapool_display
debug system display_overwrite_crash_dump
debug system overwrite_crash_dump
debug system print_stats
debug system show_crash
debug system stackpool_display

applicable models:

All models.

debug system clear_stats

This command clears T1 statistics.

parameter	definition
slot_num	Specify the slot number for the interface.

syntax:

clear_stats slot_num < 1 | 2 >

NOTE: For the OmniAccess 604 router, only slot number 1 is valid.

example:

Router/debug/system> **clear_stats**

related commands:

debug system clear_crash_dump
debug system datapool_display
debug system display_overwrite_crash_dump
debug system overwrite_crash_dump
debug system print_stats
debug system show_crash
debug system stackpool_display

applicable models:

All models.

debug system datapool_display

This command shows the network data pool allocation and usage.

syntax:

datapool_display

example:

Router/debug/system> **datapool_display**

related commands:

debug system clear_crash_dump
debug system clear_stats
debug system display_overwrite_crash_dump
debug system overwrite_crash_dump
debug system print_stats
debug system show_crash
debug system stackpool_display

applicable models:

All models.

debug system display_loaned_common_buffer

Displays information on buffers loaned from common buffer pool. This command is useful for debugging common buffer leaks.

syntax:

parameter	definition
[summary]	lists summary of the loaned buffers
[detail]	displays the packet dump of the loaned buffers
[buffer-id]	displays summary or detail of the requested buffer

debug system display_loaned_common_buffer [summary] [detail] [buffer-id]

applicable models:

All models.

debug system display_overwrite_crash_dump

This command displays the status of the overwrite crash dump.

syntax:

display_overwrite_crash_dump

example:

Router/debug/system> **display_overwrite_crash_dump**

related commands:

debug system clear_crash_dump
debug system clear_stats
debug system datapool_display
debug system overwrite_crash_dump
debug system print_stats
debug system show_crash
debug system stackpool_display

applicable models:

All models.

debug system overwrite_crash_dump

This command over writes the crash dump after a maximum of five dumps have been saved.

The default is over write.

syntax:

[no] overwrite_crash_dump

example:

Router/debug/system> **overwrite_crash_dump**

related commands:

debug system clear_crash_dump
debug system clear_stats
debug system datapool_display
debug system display_overwrite_crash_dump
debug system print_stats
debug system show_crash
debug system stackpool_display

applicable models:

All models.

debug system print_stats

This command shows T1 statistics.

syntax:

print_stats slot_num < 1 | 2 >

NOTE: For the OmniAccess 604 router, only slot number 1 is valid.

example:

Router/debug/system> **print_stats**

related commands:

debug system clear_crash_dump
debug system clear_stats
debug system datapool_display
debug system display_overwrite_crash_dump
debug system overwrite_crash_dump
debug system show_crash
debug system stackpool_display

applicable models:

All models.

debug system show_crash

This command shows system crash information.

parameter	definition
info	
detail	Detailed crash information
summary	Summary of crash information This is the default value.

syntax:

show_crash [info < detail | summary >]

example:

Router/debug/system> **show_crash detail**

related commands:

debug system clear_crash_dump
debug system clear_stats
debug system datapool_display
debug system display_overwrite_crash_dump
debug system overwrite_crash_dump
debug system print_stats
debug system stackpool_display

applicable models:

All models.

debug system stackpool_display

This command shows the network stack pool allocation and usage.

syntax:

stackpool_display

example:

Router/debug/system> **stackpool_display**

related commands:

debug system clear_crash_dump
debug system clear_stats
debug system datapool_display
debug system display_overwrite_crash_dump
debug system overwrite_crash_dump
debug system print_stats
debug system show_crash

applicable models:

All models.

debug ttcp

This command accesses next-level commands to configure and debug the ttcp client and server.

syntax:

ttcp

example:

Router/debug> **ttcp**

next-level commands

debug ttcp ttcpclient
debug ttcp ttcpserver

applicable models:

All models.

debug ttcp ttcpclient

This command debugs the ttcp client performance.

parameter	definition
dipaddress	Destination IP address
buflen	Length of the buffer in bytes The range is 8192 - 16384; the default is 8192.
nbuf	Number ot the buffer to be sent The range is 2048 - 40960; the default is 2048
bufalign	Buffer alignment The range is 8192 - 16384; the default is 16384.
bufoffset	Buffer offset The range is 0 - 1; the default is 0.
port	The port number of the server The range is 5000 - 6000; the default is 5001.
mode	Mode of the data pattern
sink	In sink mode This is the default value.
source	In source mode
delay	TCP delay or no delay The range is 0 - 1; the default is 0.

syntax:

tccpclient dipaddress < ip address > [buflen < n >] [nbuf < n >] [bufalign < n >] [bufoffset < n >] [port < n >] [mode < sink | source >] [delay < n >]

example:

Router/debug/ttcp> **ttcpclient 100.143.44.10**

related commands:

debug ttcp ttcpclient
debug ttcp ttcpserver

applicable models:

All models.

debug ttcp ttcpserver

This command debugs the ttcp server performance.

parameter	definition
buflen	The length of the buffer in bytes
	The range is 8192 - 16384; the default is 8192.
bufalign	The buffer alignment
	The range is 8192 - 16384; the default is 16384.
bufoffset	The buffer offset
	The range is 0 - 1; the default is 0.
port	The port number of the server
	The range is 5000 - 6000; the default is 5001.
mode	The mode of the data pattern
sink	In sink mode
	This is the default value.
source	In source mode
rcvwindsize	The receiving window size
	The range is 4096 - 65535; the default is 65535.

syntax:

ttcpserver [buflen < n >] [bufalign < n >] [bufoffset < n >] [port < n >] [mode < sink | source >] [rcvwindsize < n >]

example:

Router/debug/ttcp> **ttcpserver bufoffset 1**

next-level commands

debug ttcp ttcpclient

applicable models:

All models.

debug virtual-access ppp

This command provides diagnostic information on PPP processes.

syntax:

virtual-access ppp

example:

Router/debug> **virtual-access ppp**

next-level commands

debug virtual-access pppoe

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

debug virtual-access pppoe

This command provides diagnostic information on PPPoE processes.

syntax:

virtual-access pppoe

example:

Router/debug> **virtual-access pppoe**

next-level commands

debug virtual-access ppp

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

debug vlan

This command accesses next-level commands to access VLAN debug commands.

syntax:

vlan

example:

Router/debug> **vlan**

next-level commands

debug vlan vlanclear
debug vlan vlanshow
debug vlan vlantable
debug vlan vlantagtable

applicable models:

All models.

debug vlan vlanclear

This command clears VLAN statistics.

syntax:

vlanclear

example:

Router/debug/vlan> **vlanclear**

related commands:

debug vlan vlanshow
debug vlan vlantable
debug vlan vlantagtable

applicable models:

All models.

debug vlan vlanshow

This command displays VLAN statistics.

parameter	definition
verbose	Displays statistics in verbose mode

syntax:

vlanshow [verbose]

example:

Router/debug/vlan> **vlanshow**

related commands:

debug vlan vlanclear
debug vlan vlantable
debug vlan vlantagtable

applicable models:

All models.

debug vlan vlantable

This command shows the VLAN forwarding table.

syntax:

vlantable

example:

Router/debug/vlan> **vlantable**

related commands:

debug vlan vlanclear
debug vlan vlanshow
debug vlan vlantagtable

applicable models:

All models.

debug vlan vlantagtable

This command shows the VLAN tagged interfaces.

syntax:

vlantagtable

example:

Router/debug> **vlantagtable**

related commands:

debug vlan vlanclear
debug vlan vlanshow
debug vlan vlantable

applicable models:

All models.

5 DHCP

DHCP includes only configure and display commands.

NOTE: To simplify volume deployment of VoIP phones, DHCP allows an IP address to be set for an FTP server from which the phones can download their software and configuration. This option (option 66) is used to identify a TFTP server when the 'sname' field in the DHCP header has been used for DHCP options

configure interface ethernet dhcp

This command configures dynamic host configuration protocol (DHCP) relay agent for Ethernet interfaces.

syntax:

configure interface ethernet < 0 | 1 > dhcp

example:

Router/configure/interface/ethernet0> **dhcp**

next-level commands

configure interface ethernet dhcp gateway_address
configure interface ethernet dhcp server_address

applicable models:

All systems.

configure interface ethernet dhcp gateway_address

This command configures the giaddr field of dhcp relayed packets. If not specified, the giaddr field is marked with the IP address of interface.

Use the no form of this command to remove the configured gateway address.

syntax:

configure interface ethernet < 0 | 1 > dhcp gateway_address < gateway ip address >

example:

Router/configure/interface ethernet0> **dhcp gateway_address 100.4.3.2**

This example configures the gateway IP address 100.4.3.2 for dhcp relayed packets on Ethernet 0.

related commands:

configure interface ethernet dhcp server_address

applicable models:

All systems.

configure interface ethernet dhcp server_address

This command configures the dhcp server address to forward dhcp packets. It also enables/disables the dhcp relay funtionality.

Use the no form of this command to remove the configured server address.

syntax:

configure interface ethernet < 0 | 1 > dhcp server_address < server ip address >

example:

Router/configure/interface ethernet0> **dhcp server_address 120.5.4.3**

This example configures the server IP address 120.5.4.3 for dhcp relayed packets on Ethernet 0.

related commands:

configure interface ethernet dhcp gateway_address

applicable models:

All systems.

show dhcp_relay

This command displays the relay state of the Ethernet interfaces.

syntax:

display dhcp_relay

example:

Router> **show dhcp_relay**

This example shows the packet relay state for Ethernet interfaces as either enabled or disabled.

applicable models:

All systems.

6
SHOW

Use the **show** commands to show system data.

The first-level **show** commands are as follows:

show Commands

show aaa

show arp

show arp_timeout

show boot_params

show cfg_log

show configuration

show crypto

show date

show dhcp_relay

show event_logs

show fr

show ftp

show hostname

show interface

show ip

show ipmux

show ledAnalyzer

show mac

show mhu

show module

show power

show qos

show reverse_telnet

show snmp

show sntp

show system

show Commands (continued)

show telnet
show environment
show user_accounts
show users
show version
show vlanfwd
show vlanfwd macbridge
show vrrp
show whoami

show aaa radius

This command displays configured RADIUS information.

syntax:

aaa radius

example:

Router/show> **aaa radius**

screen display example

```
> show aaa radius

                RADIUS CLIENT CONFIGURATION
               ----------------------------

          Source address                : 10.1.10.0
          Primary server                : 10.1.20.10
          Secondary server              : 192.168.0.0
          Authentication port           : 1812
          Timeout in seconds            : 8
          Maximum retries               : 3
>
```

applicable models:

All models.

show arp

This command displays the current Address Resolution Protocol (ARP) table.

syntax:

arp

example:

Router/show> **arp**

The following screen capture shows destination addresses and their associated gateways, flags, usage, expire times, and interfaces.

screen display example

```
> show arp
Flags

U: Route is up
G: Route is to a gateway
H: Route is to a host
A: Route is from arp
D: Route is dynamic
P: Route is permanent
X: Route is published
C: Route is cloned
R: Route is unreachable


LINK LEVEL ARP TABLE
destination     gateway            flags Refcnt Use        Expire Interface
------------------------------------------------------------------------------
------------------------------------------------------------------------------
>
```

applicable models:

All models.

show arp_timeout

This command shows the configured value for the arp timeout period.

syntax:

arp_timeout

example:

Router> **show arp_timeout**

screen display example

```
>  show arp_timeout
Arp timeout in seconds: 1200
>
```

applicable models:

All models.

show boot_params

This command shows boot parameters for the system.

parameter	definition
slot_no	IC slot number (1 or 2)

syntax:

boot_params IC < 1 | 2 >

example:

Router/show> **boot_params 0**

screen display example

```
> show boot_params 0

FOLLOWING PARAMETERS WILL BE USED TO BOOT UP

BOARD           : NCM
HOST IP ADDRESS: 65.118.122.250
BOOTING FROM    : FLASH
BOOT FILE       : /flash1/T1000.Z
BOOT MODE       : FAST
>
```

related commands:

configure bootIC
configure bootNCM

applicable models:

All models, except OmniAccess 604.

show cfg_log

This command shows the last 10 changes to the system configuration file.

The show shows a date and time stamp for each change, the user who logged the change, and the local or network IP address the configuration file was saved to (if applicable).

syntax:

cfg_log

example:

Router/show> **cfg_log**

screen display example

```
> show cfg_log
-------------------------------------------------------------------
  No.         Time               User      IP Address     File
-------------------------------------------------------------------
   1.  05/07/2004 16:59:41            192.168.123.100
   2.  05/07/2004 16:59:41            192.168.123.100
   3.  05/07/2004 16:59:41            192.168.123.100
   4.  05/07/2004 16:59:41            192.168.123.100
   5.  05/07/2004 17:09:38      user  192.168.123.100
   6.  05/07/2004 17:27:53      user   10.121.71.220
   7.  05/07/2004 17:29:04      user   10.121.71.220
   8.  05/07/2004 17:29:35      user  192.168.120.1
   9.  05/07/2004 17:32:45      user  192.168.120.1
  10.  05/07/2004 17:42:40      user  192.168.120.1
>
```

related commands:

clear cfg_file

applicable models:

All models.

show configuration

This command accesses next-level commands for showing system configurations.

syntax:

configuration

example:

Router/show> **configuration**

next-level commands

show running-config
show startup-config

applicable models:

All models.

show crypto

This command access next-level crypto show commands.

syntax:

crypto

example:

Router1> **show crypto**

next-level commands

show crypto dynamic
show crypto ike
show crypto interfaces
show crypto ipsec
show statistics

applicable models:

All models.

show crypto dynamic clients

Displays the remote access clients who logged in successfully.

syntax:

crypto dynamic clients

example:

router> **show crypto dynamic clients**

applicable models:

All models.

show crypto dynamic ike policy [<policy-name>] [<proposal priority>] [detail]

Displays the configured dynamic IKE policies.

parameter	definition
policy-name	
all	Displays all IKE policies
policy name	Displays the specified IKE policy
proposal-priority	
1	Proposal 1
2	Proposal 2
3	Proposal 3
4	Proposal 4
5	Proposal 5
detail	Detail mode of display

syntax:

crypto dynamic ike policy <policy-name> [<proposal priority(s)>] [detail]

example:

router> show crypto ike policy toOpal 1

applicable models:

All models.

show crypto ike

This command accesses next-level commands for displaying IKE.

syntax:

ike

example:

Router1> **show crypto ike**

next-level commands

show crypto ike policy
show crypto ike sa

applicable models:

All models.

show crypto ike policy

This command displays either a specified policy or all policies in the IKE table.

parameter	definition
policy-name	
all	Displays all IKE policies
policy name	Displays the specified IKE policy
proposal-priority	
1	Proposal 1
2	Proposal 2
3	Proposal 3
4	Proposal 4
5	Proposal 5
detail	Detail mode of display

syntax:

policy policy-name [proposal-priority] [detail]

example 1:

Provides a brief display of all IKE policies in the router.

Router1> **show crypto ike policy all**

screen display example

```
>  show crypto ike policy all

Policy     Peer            Mode       Transform
------     ----            ----       ---------

>
```

example 2:

Provides a brief display of the specified IKE policy in the router.

Router1> **show crypto ike policy test**

example 3:

Provides a detailed display of the specified proposal for a specified IKE policy in the router.

Router1> **show crypto ike policy test detail**

related commands:

show crypto ike sa

applicable models:

All models.

show crypto ike sa

This command shows either all SAs for a specified policy or all SAs in the IKE table.

parameter	definition
policy-name	
all	Displays all IKE SAs
policy name	Displays all IKE SAs for this policy
detail	Detail mode of display

syntax:

sa policy-name [detail]

example 1:

Provides a brief display of all IKE SAs in the router.

Router1> **show crypto ike sa all**

screen display example

```
>  show crypto ike sa all

Policy     Peer            State      Bytes      Transform
------     ----            -----      -----      ---------


>
```

example 2:

Provides a detailed display of the specified IKE SA in the router.

Router1> **show crypto ike sa test detail**

related commands:

show crypto ike policy

applicable models:

All models.

show crypto interfaces

This command displays the network type of the interfaces.

syntax:

interfaces

example:

Router1> **show crypto interfaces**

screen display example

```
>  show crypto interfaces

Interface      Network
Name           Type
---------      -------

>
```

related commands:

show crypto ike
show crypto ipsec

applicable models:

All models.

show crypto ipsec

This command accesses next-level commands for displaying IPSec.

syntax:

ipsec

example:

Router1> **show crypto ipsec**

next-level commands

show crypto ipsec policy
show crypto ipsec sa

applicable models:

All models.

show crypto ipsec policy

This command displays either a specified policy or all policies in the IPSec table.

parameter	definition
policy-name	
all	Displays all IPSec policies
policy name	Displays the specified IPSec policy
proposal-priority	
1	Proposal 1
2	Proposal 2
3	Proposal 3
4	Proposal 4
5	Proposal 5
detail	Detail mode of display

syntax:

policy policy-name [proposal-priority] [detail]

example 1:

Provides a brief display of all IPSec policies in the router.

Router1> **show crypto ipsec policy all**

screen display example

```
>  show crypto ipsec policy all

Policy     Peer            Match                     Proto Transform
------     ----            -----                     ----- ---------

>
```

example 2:

Provides a brief display of the specified IPSec policy in the router.

Router1> **show crypto ipsec policy test**

example 3:

Provides a detailed display of the specified proposal for a specified IPSec policy in the router.

Router1> **show crypto ipsec policy test detail**

related commands:

show crypto ipsec sa

applicable models:

All models.

show crypto ipsec sa

This command displays either all SAs for a specified policy or all SAs in the IPSec table.

parameter	definition
policy-name	
all	Displays all IPSec SAs
policy name	Displays all IPSec SAs for this policy
detail	Detail mode of display

syntax:

sa policy-name [detail]

example 1:

Provides a brief display of all IPSec SAs in the router.

Router1> **show crypto ipsec sa all**

screen display example

```
>  show crypto ipsec sa all

Policy     Dest IP         Spi        Bytes      Transform
------     -------         ---        -----      ---------
>
```

example 2:

Provides a detailed display of the specified IPSec SA in the router.

Router1> **show crypto ipsec sa test detail**

related commands:

show crypto ipsec policy

applicable models:

All models.

show date

This command displays the date and time.

syntax:

date

example:

Router/show> **date**

The following screen capture shows the date and time, the number of hours and minutes ahead of or behind the UTC also appears.

screen display example

```
> show date
Local Date and Time:
SAT MAY 08 07:49:53 2004

UTC Date and Time:
SAT MAY 08 07:49:53 2004
UTC time_zone_offset:   +0:0
>
```

applicable models:

All models.

show dhcp_relay

This command displays the dhcp relay state of the Ethernet interfaces.

syntax:

dhcp_relay

example:

Router> **show dhcp_relay**

This example shows the packet relay state for Ethernet interfaces as either enabled or disabled.

screen display example

```
>  show dhcp_relay

DHCP RELAY CONFIGURATION
--------------------------
Ethernet 0: Disabled
Ethernet 1: Disabled

>
```

applicable models:

All models.

show environment

This command displays the internal operating temperature of a Router system.

Router systems are rated for operation at an ambient temperature range between 0° C to 50° C (32° F to 122° F). Two internal sensors continuously monitor the system temperature.

The following warnings are displayed if preset temperature thresholds are exceeded:

- A temperature exceeding 61° C (142° F) results in a SOFT warning.
- A temperature exceeding 71° C (160° F) results in a DANGER warning.

syntax:

temperature

example 1:

Router/show> **environment**

screen display example

```
> show  environment
Internal Unit temperature is within the recommended operating range
(NORMAL)
>
```

example 2:

For the OmniAccess 604, there are three possible messages that can be displayed, depending on the temperature of the router:

- Internal Unit temperature is within the recommended operating range (NORMAL).
- Internal Unit temperature is approaching the maximum recommended range (WARNING).
- Internal Unit temperature is exceeding the maximum recommended range (CRITICAL).

Router-OmniAccess 604> **show environment**

related commands:

test fan (not applicable to Router OmniAccess 604**)**

applicable models:

All models.

show event_logs

This command displays the system event log.

The latest 10 events appear, as shown in. To scroll through the log, 10 events at a time, type **y,** and then press **Return** at the "Display more events?" prompt. Or, to go back to the command prompt, type **n**, and press **Return**.

syntax:

event_logs

example:

Router/show> **event_logs**

The following screen capture shows the system event log. The latest 10 events appear; to scroll through the log, 10 events at a time, type **y,** and then press **Return** at the "Display more events?" prompt. Or, to go back to the command prompt, type **n**, and press **Return**.

screen display example

```
> show event_logs
5/8/2004- 7:43:55.666 -10- MB INFO MBSLOT DBGFLSH: FCLOSE :FP 8c4a0f50
Fd = 53.
5/8/2004- 7:41:40.349 -10- MB INFO MBSLOT DBGFLSH: file FOPEN  :FP
8c4a0f50 Fd =
 53. File = /flash1/system.cfg .
5/7/2004-17:34: 2.216 -10- MB INFO MBSLOT DBGFLSH: FCLOSE :FP 8c4a1490
Fd = 53.
5/7/2004-17:33:29.916 -10- MB INFO MBSLOT DBGFLSH: file FOPEN  :FP
8c4a1490 Fd =
 53. File = /flash1/system.cfg .
5/7/2004-17: 0:31.583 *3* MB ADMIN MBSLOT user logged in from CONSOLE

5/7/2004-16:59:41.683 -10- MB INFO MBSLOT DBGFLSH: FCLOSE :FP 8c6d2aa0
Fd = 53.
5/7/2004-16:59:39.216 -10- MB INFO MBSLOT DBGFLSH: file FOPEN  :FP
8c6d2aa0 Fd =
 53. File = /flash1/system.cfg .
5/7/2004-16:59:23.183 *3* MB STATUS MBSLOT temperature is NORMAL
5/7/2004-16:59:19. 16 -10- MB ADMIN MBSLOT Main program started
5/7/2004-16:59:18. 16 *3* MB STATUS MBSLOT QT1 interface card added

Display more events (Y/N) ?
```

related commands:

configure event offline
configure event online

applicable models:

All models.

show flash

This command accesses next-level commands for displaying frame relay data. This command is equivalent to **dir**.

example:

Router/show> **flash**

screen display example

```
> show flash

CONTENTS OF /flash1:

  size          date       time       name
--------       ------     ------    --------
 6467513    FEB-04-2004  13:51:22   T1000.1223.Z
 6771268    APR-01-2004  11:38:42   T1000.Z
    2401    MAY-08-2004  08:47:00   system.cfg
    1908    FEB-05-2004  07:12:30   oldsystem.cfg
 6500329    APR-01-2004  11:49:22   T1000.020404.Z

Total bytes: 19743419
Bytes Free:  12648448
>
```

next-level commands

show fr avcs
show fr cvcs
show fr invarp
show fr invarp_int
show fr lmistats
show fr pvcs
show fr vcstats

applicable models:

All models.

show fr avcs

This command shows all AVCs configured on a Router system.

This display includes the associated bundle information, virtual circuit ID (DLCI), status, policing, and IP address.

syntax:

fr avcs

example:

Router> **show fr avcs**

related commands:

show fr cvcs
show fr invarp
show fr invarp_int
show fr lmistats
show fr pvcs
show fr vcstats

applicable models:

All models.

show fr cvcs

This command shows all CVCs configured on a Router system.

This display includes the associated bundle information, virtual circuit ID (DLCI), status, policing, and IP address.

syntax:

fr cvcs

example:

Router> **show fr cvcs**

related commands:

show fr avcs
show fr invarp
show fr invarp_int
show fr lmistats
show fr pvcs
show fr vcstats

applicable models:

All models.

show fr invarp

This command displays the inverse ARP statistics for a frame relay bundle.

This command displays the DLCI numbers, local and remote IP addresses, remote netmasks, and static routing data for a particular bundle.

parameter	definition
all	Displays data for all bundles.
name	Bundle for which inverse ARP statistics will be displayed.

syntax:

fr invarp < all | *name* >

example:

Router/show> **fr invarp all**

related commands:

show fr avcs
show fr cvcs
show fr invarp_int
show fr lmistats
show fr pvcs
show fr vcstats

applicable models:

All models.

show fr invarp_int

This command displays the time interval configured for the frame relay inverse ARP timer.

syntax:

fr invarp_int

example:

Router/show> **fr invarp_int**

screen display example

```
> show fr invarp_int
 Frame relay Inverse ARP timer interval = 30 seconds
>
```

related commands:

show fr avcs
show fr cvcs
show fr invarp
show fr lmistats
show fr pvcs
show fr vcstats

applicable models:

All models.

show fr lmistats

This command displays the LMI statistics for a bundle.

Statistics include the number of received status inquiries and full status inquiries, the number of transmitted and received status inquiries, and the number of transmit-side updates.

parameter	definition
name	Bundle for which statistics will be displayed.

syntax:

fr lmistats < *name* >

example:

Router/show> **fr lmistats TNET**

related commands:

show fr avcs
show fr cvcs
show fr invarp
show fr invarp_int
show fr pvcs
show fr vcstats

applicable models:

All models.

show fr pvcs

This command displays all PVCs including their bundle name, PVC number and status, and policing (enabled or disabled) information.

If the PVC is switched to another PVC, that PVC number will be shown. If the PVC terminates at a LAN, the PVC IP address is displayed.

syntax:

fr pvcs

example:

Router/show> **fr pvcs**

related commands:

show fr avcs
show fr cvcs
show fr invarp
show fr invarp_int
show fr lmistats
show fr vcstats

applicable models:

All models.

show fr vcstats

This command displays virtual circuit statistics for a frame relay bundle.

parameter	definition
name	Bundle for which virtual circuit statistics will be displayed.
dlci	The DLCI number The range is 16 - 1022.
stat_type	
1	RX MON statistics
2	INJECT statistics
3	1490 statistics

syntax:

fr vcstats < *name* > dlci < n > [stat_type < 1 | 2 | 3 >]

example:

Router/show> **fr vcstats wan4 16**

related commands:

show fr avcs
show fr cvcs
show fr invarp
show fr invarp_int
show fr lmistats
show fr pvcs

applicable models:

All models.

show ftp

This command shows if the FTP server is enabled and who the FTP client is. The type of information displayed will depend on the user's access level.

syntax:

ftp

example:

Router> **show ftp**

screen display example

```
>  show ftp
FTP Setting:
--------------------------------------
        FTP Server:     Disabled

Allowed FTP Client:
--------------------------------------
        Username:       57sman
        Password:
>
```

applicable models:

All models.

show hostname

This command displays the system host name.
The host name is visible as the main command prompt.

syntax:

hostname

example:

Router/show> **hostname**

screen display example

```
Router> show hostname
HostName: Router
Router>
```

applicable models:

All models.

show interface

This command accesses next-level commands for displaying information about configured interfaces.

syntax:

interface

example:

Router/show> **interface**

next-level commands

show interface avc
show interface bundle
show interface bundles
show interface ethernet
show interface ethernets

applicable models:

All models.

show interface avc

This command displays information about an DTE-to-DTE MFR AVC.

Information includes the status of the AVC, status of each CVC in the AVC, individual CVC counters, aggregated AVC counters, the status reporting mode for the AVC, RED configuration and statistics, VLAN tagging on the AVC (when enabled), and IP encapsulation information.

parameter	definition
avc_name	CVC name Enter a word or name up to eight characters.
dlci	dlci of the DTE-to-DTE MFR AVC The range is 16 - 1022.

syntax:

avc avc_name < *avc name* > dlci < n >

example:

Router> **show interface avc wan05 16**

applicable models:

All models.

show interface bundle

This command displays the configuration and status for a specified bundle.

Press any key to scroll through the display, or press **q** to return to the command prompt. Repeat this command to obtain updated statistical data.

parameter	definition
name	Name of bundle to be viewed.

syntax:

bundle < *name* >

example:

Router/show/interface> **bundle Brussels**

related commands:

show interface bundles
show interface ethernet

applicable models:

All models.

show interface bundles

This command lists configured bundles.

syntax:

bundles

example:

Router/show/interface> **bundles**

The following (screen display example) shows a typical display of configured bundles and their bandwidths, descriptions, contact names, and status.

screen display example

```
> show interface bundles
bundle   bw(kbps) description              contact         stat IType Link
------   -------- -----------              -------         ---- ----- ----
dallas   0        -                        -               down NONE


Link Usage Summary:
-------------------
T1     NONE

>
```

related commands:

show interface bundle
show interface ethernet

applicable models:

All models.

show interface ethernet

This command displays configuration and status for a specified Ethernet port.

Press any key to scroll through the display, or press **q** to return to the command prompt. Repeat this command to obtain updated statistical data.

parameter	definition
port number	Ethernet port to be viewed (0 or 1).

syntax:

ethernet < 0 | 1 >

example:

Router/show/interface> **ethernet 0**

To clear the Ethernet port display counters, use the **clear counters Ethernet** command.

screen display example

```
> show interface ethernet 0

ethernet 0
ipaddr       10.10.1.1
netmask      255.255.255.0
description  -
status       down, operationally down
configured   auto
  speed       -
  mode        -
actual
  speed      100
  mode       half_duplex
mtu          1500

ethernet0 (unit number 0)
Type: ETHERNET (802.3)
Flags: (0x807c203)  UP, MULTICAST-ROUTE
Internet Address: 10.10.1.1
Internet Netmask: 255.255.255.0
Internet Broadcast: 10.10.1.255
Maximum Transfer Unit: 1500 bytes
Mac Address:  00:00:23:00:60:00

port counters since last boot/clear
  Bytes Rx                            0  Bytes Tx                            0

  Packets Rx                          0  Packets Tx                          0

  Runts Rx                            0  Collisions                          0

  Babbels Rx                          0  Late Collisions                     0

  Err Packets Rx                      0  Up/Down States (Phys)               0

  Up/Down States (Admin)              2

port counters for the last five minutes
  Bytes Rx                            0  Bytes Tx                            0

  Packets Rx                          0  Packets Tx                          0

  Runts Rx                            0  Collisions                          0

  Babbels Rx                          0  Late Collisions                     0

  Err Packets Rx                      0  Up/Down States (Phys)               0

  Up/Down States (Admin)              0
>
```

related commands:

clear counters Ethernet

applicable models:

All models.

show interface ethernets

This command displays Ethernet port configuration and status.

This display shows the IP address, subnet mask, user-defined description, status (up = active; down = shut down), speed (10 or 100 MHz), and operating mode (half-duplex or full-duplex) for each port.

syntax:

ethernets

example:

Router/show/interface> **ethernets**

screen display example

```
> show interface ethernets


ethernet 0
ipaddr       10.10.1.1
netmask      255.255.255.0
description  -
status       down, operationally down
configured   auto
  speed       -
  mode        -
actual
  speed      100
  mode       half_duplex
mtu          1500


ethernet 1
ipaddr       192.168.120.1
netmask      255.255.255.0
description  -
status       down, operationally down
configured   auto
  speed       -
  mode        -
actual
  speed      100
  mode       half_duplex
mtu          1500
>
```

related commands:

show interface ethernet
show interface bundle

applicable models:

All models.

show interface virtual-access

This command accesses PPPoE interface information.

syntax:

interface virtual-access

example:

Router/show> **interface virtual-access**

next-level commands

show virtual-access

applicable models:

OmniAccess 601, OmniAccess 602, and OmniAccess 604

show ip

This command accesses next-level commands for displaying IP data.

syntax:

ip

example:

Router/show> **ip**

next-level commands

show ip dns
show ip hosts
show ip interfaces
show ip nat
show ip access-list
show ip routes

applicable models:

All models.

show ip dhcps address_pools

This command displays the DHCP address pools.

syntax:

address_pools

example:

Router/show/ip> **dhcps address_pools**

related commands:

show ip dhcps bindings
show ip dhcps configuration
show ip dhcps interfaces
show ip dhcps statistics

applicable models:

All models.

show ip dhcps bindings

This command displays the DHCP server bindings.

syntax:

bindings

example:

Router/show/ip> **dhcps bindings**

related commands:

show ip dhcps address_pools
show ip dhcps configuration
show ip dhcps interfaces
show ip dhcps statistics

applicable models:

All models.

show ip dhcps configuration

This command displays the DHCP configuration.

syntax:

configuration

example:

Router/show/ip> **dhcps configuration**

related commands:

show ip dhcps bindings
show ip dhcps address_pools
show ip dhcps interfaces
show ip dhcps statistics

applicable models:

All models.

show ip dhcps interfaces

This command displays the DHCP interfaces.

syntax:

bindings

example:

Router/show/ip> **dhcps interfaces**

related commands:

show ip dhcps bindings
show ip dhcps address_pools
show ip dhcps configuration
show ip dhcps statistics

applicable models:

All models.

show ip dhcps statistics

This command displays the DHCP server statistics.

syntax:

statistics

example:

Router/show/ip> **dhcps statistics**

related commands:

show ip dhcps bindings
show ip dhcps address_pools
show ip dhcps configuration
show ip dhcps interfaces

applicable models:

All models.

show ip dns

This command displays the DNS name.

syntax:

dns

example:

Router/show/ip> **dns**

related commands:

show ip hosts

applicable models:

All models.

show ip hosts

This command displays a list of configured hosts, along with their IP addresses.

syntax:

hosts

example:

Router/show/ip> **hosts**

screen display example

```
> show ip hosts
hostname         inet address       aliases
--------         ------------       -------
localhost        127.0.0.1
user             192.168.120.1
user              10.10.1.1
>
```

related commands:

configure ip host_add

applicable models:

All models.

show ip interfaces

This command displays Ethernet port IP routing data.

syntax:

interfaces [brief]

example:

Router/show/ip> **interfaces**

screen display examples

```
> show ip interfaces

ethernet0.1 (unit number 3)
Type: ETHERNET (802.3)
Flags: (0x7c243)  UP, RUNNING, MULTICAST-ROUTE
Internet Address: 101.111.111.111
Internet Netmask: 255.255.255.0
Internet Broadcast: 101.111.111.255
Maximum Transfer Unit: 1500 bytes
Mac Address:  00:00:23:00:7e:03
....

> show ip interfaces interface ethernet1.1

ethernet1.1 (vlan:4095) (unit number 4)
Type: ETHERNET (802.1q)
Flags: (0x7c203)  UP, MULTICAST-ROUTE
Internet Address: 200.200.200.200
Internet Netmask: 255.255.255.0
Internet Broadcast: 200.200.200.255
Maximum Transfer Unit: 1500 bytes
Mac Address:  00:00:23:00:7e:01


> show ip interfaces brief
Interface                 Type             IP-Address/Mask    Status
null                      NULL             127.1.0.2/24       Up
ethernet0.1               ETHERNET (802.3) 101.111.111.111/24 Up
ethernet1.1 (vlan:4095)   ETHERNET (802.1q) 200.200.200.200/24 Admin. down
ethernet1.2047 (vlan:3999) ETHERNET (802.1q) 122.122.122.122/18 Down
wanintf (dlci:1000)       PT2PT            100.1.1.1/24       Down
myloopbk                  S/W LOOPBACK     223.234.233.222/20 Up


> show ip interfaces interface ethernet1.1 brief
Interface                  Type             IP-Address/Mask    Status
ethernet1.1 (vlan:4095)    ETHERNET (802.1q) 200.200.200.200/24 Down
>
```

related commands:

show interface ethernet
show interface ethernets
configure interface ethernet

applicable models:

All models.

show ip interface

This command displays Ethernet port IP routing data for the specified interface.

syntax:

interface <interface name>

example:

Router/show/ip> **interface ethernet0**

related commands:

show interface ethernet
show interface ethernets
configure interface ethernet

applicable models:

All models.

show ip nat

This command accesses next-level commands for displaying network address translation (NAT) data.

syntax:

nat

example:

Router/show/ip> **nat**

next-level commands

show ip nat address_pool
show ip nat all
show ip nat configuration
show ip nat statistics
show ip nat translations

applicable models:

All models.

show ip nat address_pool

This command displays information about NAT address pools.

syntax:

address_pool [name]

example 1:

This example displays information about all address pools.

Router/show/ip/nat> **address_pool**

example 2:

This example shows information only about pool1.

Router/show/ip/nat> **address_pool pool1**

applicable models:

All models.

show ip nat all

This command displays a summary of all stored NAT data.

syntax:

all

example:

Router/show/ip/nat> **all**

The following screen display shows a typical display showing local IP addresses and ports, global IP addresses and ports, and their associated protocols.

applicable models:

All models.

show ip nat configuration

This command shows global NAT data or NAT data for a specified interface.

parameter	definition
interface	
ethernet0	Ethernet 0
ethernet1	Ethernet 1
bundle_name	Specific WAN bundle name
bundle_name:pvc _number	Bundle name:PVC number
global	Global NAT

syntax:

interface < ethernet0 | ethernet1 | bundle_name | bundle_name : pvc_number > < global >

example 1:

This example displays NAT information for the Ethernet 0 interface.

Router/show/ip/nat> **configuration wan1**

example 2:

This example displays global NAT information.

Router/show/ip/nat> **configuration global**

applicable models:

All models.

show ip nat statistics

This command shows global NAT statistics or NAT statistics for a specified interface.

parameter	definition
interface	
ethernet0	Ethernet 0
ethernet1	Ethernet 1
bundle_name	Specific WAN bundle name
bundle_name:pvc _number	Bundle name:PVC number
global	Global NAT

syntax:

statistics interface < ethernet0 | ethernet1 | bundle_name | bundle_name : pvc_number > < global >

example 1:

Router/show/ip/nat> **statistics wan1**

screen display example

```
>  show ip nat statistics


Network Address Translation Statistics - Global

Static Address/Port Translation  - ENABLED
 Static Addresses: 1
 Static Ports: 0

Dynamic Port Translation - ENABLED
 Dynamic Ports: 0

Dynamic Address Translation - DISABLED
 Dynamic Addresses: 0

Total Translation Entries: 1

Outgoing Translations -
 Hits: 0  Misses: 0

Incoming Translations -
 Hits: 0  Misses: 0

Pass Thru -
 Outgoing: 0  Incoming: 0

Inactivity Timers -
 TCP: 7200 seconds
 UDP/ICMP: 60 seconds
 Dynamic Address Entries: 3600 seconds


>
```

example 2:

This example shows global NAT statistics.

Router/show/ip/nat**> statistics global**

applicable models:

All models.

show ip nat translations

This command shows the network address translations that occur globally or for a specified interface.

parameter	definition
interface	
ethernet0	Ethernet 0
ethernet1	Ethernet 1
bundle_name	Specific WAN bundle name
bundle_name:pvc _number	Bundle name:PVC number

syntax:

translations interface < ethernet0 | ethernet1 | bundle_name | bundle_name : pvc_number >

example 1:

This example shows NAT data for a specific interface.

Router/show/ip/nat> **translations ethernet0**

screen display example

```
> show ip nat translations ethernet0


Network Address Translation Table - Global: All Entries

Local IP Address     Local Port    Global IP Address    Global Port    Protocol
----------------     ----------    -----------------    -----------
--------
192.168.123.1        *             10.121.71.217        *              STATIC



>
```

example 2:

This example shows global NAT data.

Router/show/ip/nat> **translations global**

applicable models:

All models.

show ip access-list

This command accesses next-level commands for displaying IP packet filtering data.

syntax:

access-list

example:

Router/show> **ip access-list**

next-level commands

show ip access-list filter-list
show ip access-list rules
show ip access-list statistics

applicable models:

All models.

show ip access-list filter-list

This command displays IP packet filter list rules, or all filtering rule sets currently in the Router system.

parameter	definition
rulelist_name	Name of the filter set for which rules will be displayed.

syntax:

filter-list rulelist_name < name >

example:

Router/show/ip/access-list> **filter-list Filter01**

related commands:

show ip access-list rules
show ip access-list statistics

applicable models:

All models.

show ip access-lists-rules

This command shows the IP filtering sets applied to an Ethernet port, a bundle, a PVC, or all interfaces. The names of the filter sets are given for both packet transmission directions.

parameter	definition
interface	
ethernet0	Ethernet 0
ethernet1	Ethernet 1
bundle_name	Bundle name
bundle_name:pvc_no	Bundle name:PVC number
all	All interfaces

syntax:

access-list rules interface < ethernet0 | ethernet1 | bundle_name | bundle_name : pvc_no | all >

example:

Router/show/ip> **access-list rules wan1:3**

related commands:

show ip access-list filter-list
show ip filter-list statistics

applicable models:

All models.

show ip access-list statistics

This command shows IP filter statistics for an Ethernet port, a bundle, a PVC, or all interfaces.

parameter	definition
interface	
ethernet0	Ethernet 0
ethernet1	Ethernet 1
bundle_name	Bundle name
bundle_name:pvc_no	Bundle name:PVC number
all	All interfaces

syntax:

access-list statistics interface < ethernet0 | ethernet1 | bundle_name | bundle_name : pvc_no | all >

example:

Router/show/ip> **access-list statistics ethernet0**

related commands:

show ip access-list filter-list
show ip filter-list rules

applicable models:

All models.

show ip routes

This command displays IP routing information.

parameter	definition
network	Network IP address
mask	Net mask address
protocol	
all	All protocols
bgp	Border Gateway protocol
connected	Connected routes
ospf	Open Shortest Path First protocol
rip	Routing Information protocol
static	Static routes
database	Route database
fib	FIB routes
summary	
summary	Summary of all routes
options	output modifiers
begin	Begin with the route that matches the network/mask wildcard
exact	Exact match with network/mask.
exclude	Exclude routes that match the network/mask wildcard
include	Include routes that match the network/mask wildcard

syntax:

routes [network < ip address >] [mask < net mask >] [protocol < all | bgp | connected | ospf | rip | static >] [database < fib >] [summary < summary >] [options < begin | exact | exclude | include >]

example 1:

> **show ip routes**

If used with database option "fib," then routes in FIB (forwarding table) are displayed.

example 2:

> **show ip routes summary**

This command provides the distribution of routes in the RIB (routing table) or FIB based on the source/owner of the route (e.g., Static, BGP, OPSPF, etc.). If used with database option "fib," then the display will show the route summary of FIB.

screen display example

```
>  show ip routes summary
IP Routing Table Summary
Route Source       Number of Routes
------------       ----------------
connected       :  0
static          :  0
aggregate       :  0
BGP             :  0
RIP             :  0
OSPF            :  0

Total           :  0

>
```

example 3:

> **show ip routes network 192.168.29.107**

This command provides the best match route for the destination x.x.x.x from the RIB. The database option "fib" is not supported.

example 4:

> **show ip route network 11.0.0.0 mask 255.0.0.0**

This command provides the matching routes to the network and mask combination. The implicit output modifier "included" can be used with this command. The database options "rib" and"fib" can also be applied.

example 5:

> **show ip routes network 11.0.0.0 mask 255.0.0.0 exclude**

The output modifier "exclude" will force all the matching routes to the network and mask conbination to not be listed in the displayed route information. The database options "rib" or "fib" can also be applied.

example 6:

> **show ip routes network 11.0.0.0 mask 255.0.0.0 begin**

When the output modifier "begin" is used, the displayed route information will begin with the first route that matches the network and masks combination. The database options "rib" or "fib" can also be applied.

applicable models:

All models.

show ip rtp

This command accesses next-level commands for displaying RTP information.

syntax:

show ip rtp

example:

Router> **show ip rtp**

applicable models:

All models.

show ip rtp configuration

This command shows the RTP configuration for the specified bundle.

syntax:

configuration interface < bundle name >

example:

Router/show/ip/rtp> **configuration wan1**

applicable models:

All models.

show ip rtp statistics

This command shows the RTP statistics for the specified bundle.

syntax:

statistics interface < bundle name >

example:

Router/show/ip/rtp> **statistics wan1**

applicable models:

All models.

show ipmux

This command accesses next-level commands ipmux display commands.

syntax:

ipmux

example:

Router> **show ipmux**

next-level commands

show acroutes
show interfaces
show routes

applicable models:

All models.

show ipmux acroutes

This command shows autoconfigured ipmux routes.

syntax:

acroutes

example:

Router> **show ipmux acroutes**

related commands:

show ipmux interfaces
show ipmux routes

applicable models:

All models.

show ipmux interfaces

This command shows configured interface information.

parameter	definition
interface	Interface name or type The default is all.

syntax:

interfaces [interface < name >]

example:

Router> **show ipmux interfaces**

related commands:

show ipmux acroutes
show ipmux routes

applicable models:

All models.

show ipmux routes

This command shows information about configured ipmux routes.

parameter	definition
destination	Destination IP address The default is all.

syntax:

routes [destination < ip address >]

example:

Router> **show ipmux routes**

related commands:

show ipmux acroutes
show ipmux interfaces

applicable models:

All models.

show mac

This command displays the MAC address assigned to a Router system.

syntax:

mac

example:

Router> **show mac**

screen display example

```
> show mac
Starting MAC = 00:00:23:00:00:00
MAC Range = 512
>
```

applicable models:

All models.

show mhu

This command displays the current MHU configuration and a table of the host entries.

syntax:

show mhu

example:

Router> **show mhu**

If the host IP address matches the DHCP IP address, the client is a DHCP client. If the host IP address does not match the DHCP address, the client is a static client.

applicable models:

All models.

show module

This command accesses next-level commands for displaying interface card alarms, configurations, and statistics.

syntax:

module

example:

Router/show> **module**

next-level commands

show module alarms
show module ansistats
show module attstats
show module configuration
show module hssi_leads
show module ietfstats
show module t1_remote_mapping
show module test
show module thresholds
show module userstats

applicable models:

All models.

show module alarms

This command accesses next-level commands for displaying interface alarms.

syntax:

alarms

example:

Router/show/module> **alarms**

In alarm displays, Off = No Alarm, On = Alarm in Progress.

next-level commands

show module alarms ct3
show module alarms e1
show module alarms t1
show module alarms t3
show module alarms ussi

applicable models:

All models.

show module alarms e1

This command displays the alarms detected on an E1 interface.

Also shows individual user threshold alarms for all user statistics currently configured with alarm thresholds. To see the current user alarm threshold settings, use the **show module thresholds e1** command.

To display the alarms in real-time, specify the desired refresh interval in minutes when entering this command. To return to the system command prompt, type **q**.

parameter	definition
e1	E1 link The range is 1 - 16, depending on the Router system.
refresh_interval	How often, in minutes, the alarm display will be updated (optional entry; minimum is 1 minute). If you do not specify an interval, you must repeat the command to update the alarm display.

syntax:

e1 < n > [refresh_interval < n >]

example:

RouterE/show/module/alarms> **e1 1 refresh_interval 1**

related commands:

show module thresholds e1

applicable models:

All E1 models.

show module alarms t1

This command displays the alarms detected on a T1 interface.

Also, displays individual user threshold alarms for all user statistics currently configured with alarm thresholds. To view the current user alarm threshold settings, use the **show module thresholds t1** command.

To display the alarms in real-time, specify the desired refresh interval in minutes when entering this command. To return to the system command prompt, type **q**.

parameter	definition
t1	T1 link The range is 1 - 16, depending on the Router system.
refresh_interval	How often, in minutes, the alarm display will be updated (optional entry; minimum is 1 minute). If you do not specify an interval, you must repeat the command to update the alarm display.

syntax:

t1 < n > [refresh_interval < n >]

example:

Router/show/module/alarms> **t1 1**

In alarm displays, Off = No Alarm, On = Alarm in Progress

related commands:

show module thresholds t1

applicable models:

All T1 models.

show module ansistats

This command accesses next-level commands for displaying ANSI performance statistics.

syntax:

ansistats

example:

Router/show/module> **ansistats**

next-level commands

show module ansistats ct3
show module ansistats t1
show module ansistats t3

applicable models:

All models.

show module ansistats t1

This command displays ANSI statistics for a T1 interface.

The display shows the current 15-minute interval statistics, the elapsed time in the current sampling interval, and the total counts for the past 8 hours. These statistics may be cleared only by the carrier by FDL requests from the remote end.

Statistics for multiple 15-minute intervals may be displayed by specifying the number of intervals (1 to 96) you wish to view.

parameter	definition
t1	T1 link The range is 1 - 16, depending on the Router system.
interval_range	Range of 15-minute intervals for which you wish to display a summary of statistics (1 - 96).

syntax:

t1 < n > [interval_range < n >]

example:

Router/show/module/ansistats> **t1 4**

screen display example

```
> show module ansistats t1 4 1

System Time : SAT MAY 08 09:17:45 2004
ANSI Archive Statistics
-----------------------

Archive No 1 Start Time: SAT MAY 08 09:00:55 2004
-------------------------------------------------
INTR  T1No      CV     ES    SES    SAS    CSS    UAS       (l)CV  (l)ES (l)SES
----  ----      --     --    ---    ---    ---    ---       -----   ----  -----
 1     4         0      0      0      0      0    900           0     0        0
----  ----      --     --    ---    ---    ---    ---       -----   ----  -----
>
```

applicable models:

All T1 models.

sshow module attstats

This command accesses next-level commands for displaying AT&T statistics.

syntax:

attstats

example:

Router/show/module> **attstats**

next-level commands

show module attstats ct3
show module attstats t1
show module attstats t3

applicable models:

All models.

show module attstats t1

This command displays AT&T statistics for a T1 interface.

Statistics for multiple 15-minute intervals may be displayed by specifying the number of intervals (1 to 96) you wish to view.

parameter	definition
t1	T1 link The range is 1 - 16, depending on the Router system
interval_range	Range of 15-minute intervals for which you wish to display a summary of statistics (1 - 96).

syntax:

attstats t1 < 1 - 16 > [interval_range < n >]

example:

Router/show/module/attstats> **t1 4**

screen display example

```
> show module attstats t1 2

T1 Number 2:
System Time : SAT MAY 08 09:20:05 2004
Telco ATT Current Statistics(244 seconds elapsed)
-----------------------------------------
        EEV         ES        UAS        BES        SES       LOFC        CCS
        ---         --        ---        ---        ---       ----        ---
          0          0        244          0          0          0          0

Telco ATT Total Statistics:
-------------------------

        EEV         ES        UAS        BES        SES       LOFC        CCS
        ---         --        ---        ---        ---       ----        ---
          0          0      56043          0          0          0          0


>
```

applicable models:

All T1 models.

show module configuration

This command accesses next-level commands for displaying the configuration and status of the T1 and E1 interfaces.

syntax:

configuration

example:

Router/show/module> **configuration**

next-level commands

show module configuration all
show module configuration ct3
show module configuration e1
show module configuration t1
show module configuration t3
show module configuration ussi

applicable models:

All models.

show module configuration all

This command shows the configuration for all T1 or E1 ports.

syntax:

all

example:

Router> **show module configuration all**

screen display example

```
>  show module configuration all

T1 #      Framing   Coding    ClkSrc    LBO-CableLength   State     Alarm
-------------------------------------------------------------------------
1         esf       b8zs      int       csu/0db           down      RLOS
2         esf       b8zs      int       csu/0db           down      RLOS
3         esf       b8zs      int       csu/0db           down      RLOS
4         esf       b8zs      int       csu/0db           down      RLOS
>
```

applicable models:

All models.

show module configuration e1

This command displays the configuration and status of a E1 interface.

parameter	definition
e1	E1 link The range is 1 - 16, depending on the Router system.

syntax:

e1 < n >

example:

RouterE/display/module/configuration> **e1 1**

applicable models:

All E1 models.

show module configuration t1

This command displays the configuration and status of a T1 interface.

parameter	definition
t1	T1 link The range is 1 - 16, depending on the Router system.

syntax:

t1 < n >

example:

Router/show/module/configuration> **t1 1**

screen display example

```
>  show module configuration t1 1
T1 1 is ENABLED
Alarm Hierarchy: TRUE,
Yellow Alarm: DISABLE
Framing:ESF, LineCode:B8ZS, ClockSource: INT, LineMode:CSU, LBO:0 db
FDL: ANSI Unit Protocol enabled ,ATT Unit Protocol enabled ,
CsuDsuType:  CSU & DSU , Loopback Framing (In-band): Overwrite,
CIRCUIT-ID : Not Configured ,CONTACT-INFO : Not Configured ,
DESCRIPTION : Not Configured , LINK NAME : Not Configured ,

Line Status:
   RLOS: ON    RAIS:OFF    RLOF:OFF    RRAI:OFF    TAIS:OFF
   TRAI:OFF  TLnCod:OFF  TPlCod:OFF TRstCod:OFF   TPtrn:OFF
   Loop:OFF    LORC:OFF

Timeslot Map:
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 ___________________________________________________________________________
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__|

>
```

applicable models:

All T1 models.

show module ietfstats

This command accesses next-level commands for displaying IETF statistics.

syntax:

ietfstats

example:

Router/show/module> **ietfstats**

next-level commands

show module ietfstats ct3
show module ietfstats e1
show module ietfstats t1
show module ietfstats t3

applicable models:

All models.

show module ietfstats e1

This command displays the IETF performance statistics for an E1 interface.

Statistics for multiple 15-minute intervals may be displayed by specifying the number of intervals (1 to 32) you wish to view.

parameter	definition
e1_no	E1 link The range is 1 - 16, depending on the Router system.
interval_range	Range of 15-minute intervals for which you want to display a summary of statistics The range is 1 - 96.

syntax:

e1 e1_no < n > [interval_range < n >]

example:

RouterE/show/module/itutstats> **e1 1 interval_range 1-5**

related commands:

show module itutstats e1

applicable models:

All E1 models.

show module ietfstats t1

This command displays IETF performance statistics for a T1 interface.

Statistics for multiple 15-minute intervals may be displayed by specifying the number of intervals (1 to 96) you wish to view.

parameter	definition
t1_no	T1 link The range is 1 - 16, depending on the Router system.
interval_range	Range of 15-minute intervals for which you want to display a summary of statistics The range is 1 - 96.

syntax:

t1 t1_no < n > [interval_range < n >]

example:

Router/show/module/ietfstats> **t1 1 interval_range 1-5**

screen display example

```
>  show module ietfstats t1 1 interval_range 1-5

System Time : SAT MAY 08 09:33:44 2004
IETF Archive Statistics
-----------------------

Archive No 1 Start Time: SAT MAY 08 09:16:40 2004
-------------------------------------------------
INTR  T1No     ES   SES   SEFS   UAS   CSS   PCV   LES  BES   DM   LCV
----  ----     --   ---   ----   ---   ---   ---   ---  ---   --   ---
   1     1      0     0     0   900     0     0     0    0    0    0
----  ----     --   ---   ----   ---   ---   ---   ---  ---   --   ---
   2     1      0     0     0   900     0     0     0    0    0    0
----  ----     --   ---   ----   ---   ---   ---   ---  ---   --   ---
   3     1      0     0     0   900     0     0     0    0    0    0
----  ----     --   ---   ----   ---   ---   ---   ---  ---   --   ---
   4     1      0     0     0   900     0     0     0    0    0    0
----  ----     --   ---   ----   ---   ---   ---   ---  ---   --   ---

Archive No 5 Start Time: SAT MAY 08 08:16:40 2004
-------------------------------------------------
INTR  T1No     ES   SES   SEFS   UAS   CSS   PCV   LES  BES   DM   LCV
----  ----     --   ---   ----   ---   ---   ---   ---  ---   --   ---
   5     1      0     0     0   900     0     0     0    0    0    0
----  ----     --   ---   ----   ---   ---   ---   ---  ---   --   ---
>
```

applicable models:

All T1 models.

show module itutstats

This command accesses next-level commands that display the International Telecommunication Union statistics for an E1 port.

syntax:

itutstats

example:

AR1208-E/display/module> **itutstats**

related commands:

show module itutstats e1

applicable models:

All models.

show module itutstats e1

This command displays the International Telecommunication Union statistics for an E1 port.

parameter	definition
e1_no	E1 link (1 - 16, depending on the Router system model).
interval_range	Range of 15-minute intervals for which a summary of statistics will be displayed. The range is 1 - 96.

syntax:

itutstats e1 e1_no [interval_range < n >]

example:

RouterE/show/module> **itutstats e1 1**

related commands:

show module ietfstats e1

applicable models:

All E1 models.

show module t1_remote_mapping

This command displays T1 assignments for two Router systems that are directly connected. Depending on the system, the display will show T1s 1-16 or 1-28.

syntax:

t1_remote_mapping

example:

Router/show/module> **t1_remote_mapping**

NOTE: This command is functional only when a Router system is on both ends. Also, this command cannot resolve T1 port numbers if multiple remote Router systems are connected.

applicable models:

All T1 models.

show module test

This command accesses next-level commands for displaying the current status of an interface test on a Router system.

syntax:

test

example:

Router/show/module> **test**

() and () summarize BERT and loopback information for E1 testing.

E1 BERT Test Criteria

Display	Description
Test Type	BERT (test in progress).
	NONE (no test in progress).
Pattern	Type of BERT pattern being sent.
Status	SEARCHING (receiver is trying to lock onto a BERT pattern from the far end).
	LOCKED (receiver has recognized the BERT pattern and is locked onto it).
	RELOCKED (receiver has lost synchronization to the BERT pattern, then locked onto it again).
Locked Seconds	Number of seconds the BERT test has been in the locked state.
Pattern Loss Count	Number of times the receiver could not lock onto a BERT pattern.
Bit Error Count	Number of bit errors detected in the incoming BERT pattern.
Configured Time	Minutes and seconds for which the current BERT test is set to run.
Elapsed Time	Minutes and seconds elapsed for the current BERT test.

E1 Loopback Test Criteria

Display	Description
Loop Type	Type of loopback in progress (line, payload, or none).
Local Loop State	Current loopback state (looped up or looped down).
Remote Status	PASSED (loop-up was successful at far end).
	IDLE (loop-down was successful).
Loop Type	Type of loopback operation (loop-up or loop-down).
Loop Code	Type of loop-up code sent to the far end (ANSI FDL or Inband).

next-level commands

show module test ct3
show module test e1

show module test t1
show module test t3
show module test ussi

applicable models:

All models.

show module test e1

This command displays the test status of an E1 interface.

This display is similar to when a BERT test is in progress on an E1 channel.

parameter	definition
e1	E1 link The range is 1 - 16, depending on the Router system.

syntax:

e1 < n >

example:

RouterE/show/module/test> **e1 1**

related commands:

show module test ussi

applicable models:

All E1 models.

show module test t1

This command displays the test status of a T1 interface.

If no test is in progress, the message "Currently no tests are being run" appears. This message is followed by the status of the last test or loopback operation performed.

parameter	definition
t1	T1 link The range is 1 - 16, depending on the Router system.

syntax:

t1 < n >

example:

Router/show/module/test> **t1 1**

screen display example

```
>  show module test t1 1

Currently no tests are being run

Last Test Status:
No test has been run on this interface since reboot
>
```

related commands:

show module test ct3
show module test e1
show module test t3
show module test ussi

applicable models:

All T1 models.

show module thresholds

This command accesses next-level commands for displaying alarm thresholds.

These displays show the threshold types, along with the user-defined rising and falling alarm thresholds, sampling intervals, and sample types for each threshold.

syntax:

thresholds

example:

Router/show/module> **thresholds**

related commands:

show module thresholds ct3
show module thresholds e1
show module thresholds t1
show module thresholds t3

applicable models:

All models.

show module thresholds e1

This command displays the alarm threshold settings for an E1 interface.

parameter	definition
e1	E1 link The range is 1 - 16, depending on the Router system.

syntax:

e1 < n >

example:

RouterE/show/module/thresholds> **e1 1**

applicable models:

All E1 models.

show module thresholds t1

This command displays the alarm threshold settings for a T1 interface.

parameter	definition
t1	T1 link The range is 1 - 16, depending on the Router system.

syntax:

t1 < n >

example:

Router/show/module/thresholds> **t1 1**

screen display example

```
>  show module thresholds t1 1


THRESHOLD  VARIABLE    RISING      FALLING   INTERVAL  SAMPLETYPE
---------  --------    ------      -------   --------  ----------
>
```

applicable models:

All T1 models.

show module userstats

This command accesses next-level commands for displaying user statistics.

Statistics for multiple 15-minute intervals may be displayed by specifying the number of intervals (1 to 32) that you wish to view.

To clear these statistics after viewing them, use the **clear module t1_userstats** command.

syntax:

userstats

example:

Router/show/module> **userstats**

next-level commands

show module userstats ct3
show module userstats e1
show module userstats t1
show module userstats t3
show module userstats ussi

applicable models:

All models.

show module userstats e1

This command displays performance statistics for an E1 interface.

Statistics for multiple 15-minute intervals may be displayed by specifying the number of intervals (1 to 32) that you wish to view.

To clear these statistics after viewing them, use the **clear module e1_userstats** command.

parameter	definition
e1_no	E1 link The range is 1 - 16, depending on the Router system.
interval_range	Range of 15-minute intervals for which you want to display a summary of statistics The range is 1 - 96.

syntax:

e1 e1_no < n > [interval_range < n >]

example:

RouterE/show/userstats> **e1 2 interval_range 1-5**

applicable models:

All E1 models.

show module userstats t1

This command displays performance statistics for a T1 interface.

Statistics for multiple 15-minute intervals may be displayed by specifying the number of intervals (1 to 96) you wish to view.

To clear these statistics after viewing them, use the **clear module t1_userstats** command.

parameter	definition
t1	T1 link The range is 1 - 16, depending on the Router system.
interval_range	Range of 15-minute intervals for which you wish to display a summary of statistics (1 - 96).

syntax:

t1 < 1 - 16 > [interval_range < n >]

example:

Router/show/module/userstats> **t1 2 interval_range 1-3**

screen display example

```
>  show module userstats t1 2 interval_range 1-3

System Time : SAT MAY 08 09:40:54 2004
User Archive Statistics
-----------------------

Archive No 1 Start Time: SAT MAY 08 09:17:00 2004
-------------------------------------------------
INTR  T1No    EEV  ES   UAS   BES   SES   LOFC  CSS   BPV   OOF   CRC
----  ----    ---  --   ---   ---   ---   ----  ---   ---   ---   ---
 1     2      0     0   900     0     0     0     0     0     0     0
----  ----    ---  --   ---   ---   ---   ----  ---   ---   ---   ---
 2     2      0     0   900     0     0     0     0     0     0     0
----  ----    ---  --   ---   ---   ---   ----  ---   ---   ---   ---
 3     2      0     0   900     0     0     0     0     0     0     0
----  ----    ---  --   ---   ---   ---   ----  ---   ---   ---   ---
>
```

applicable models:

All T1 models.

show power

This command displays the type of power supply (AC, single DC, or dual DC) installed in the system.

syntax:

power

example:

Router/show> **power**

screen display example

```
> show power
power_display: AC power supply on this product!
>
```

applicable models:

All models.

show qos

This command accesses next-level commands for displaying QoS configurations and available QoS class templates.

syntax:

qos

example:

Router> **show qos**

next-level commands

show qos bundle

applicable models:

All models.

show qos bundle

This command displays either a specific class or all QoS classes configured on a specified bundle.

Specifying a class name gives detailed information about the class. Specifying only the bundle name displays all the classes configured on the bundle.

parameter	definition
name	Name of bundle containing QoS classes to be displayed.
class	Individual QoS class to be displayed The default is all.

syntax:

bundle bundle < *name* > [class < *class* >]

example:

Router/show/qos> **bundle wan1**

```
> show qos bundle wan1

Interface: Bundle wan1 (Bandwidth = 3072Kbps)

Interface Outbound Configuration & Statistics
---------------------------------------------
CBQ: on  Policing: off  MON: off
+------------------+------+------+------+-------+-------+----------+----------
Traffic Class       CBQ-CR CBQ-BR Police Avg Out  Avg In    Packets    Packets
                    (kbps) (kbps) (kbps)  (kbps)  (kbps)      Fwded    Dropped
+------------------+------+------+------+-------+-------+----------+----------
s1                     500   3072      -  1781.2  2666.6       4451       2220
  s1-def               100   3072    800   891.6  1333.1       2229       1107
  s1-web               100   3072    900   889.5  1333.4       2222       1113
def-o                  500   3072   1333  1290.7  1333.4       3230        109

Interface Inbound Configuration & Statistics
--------------------------------------------
Policing: on  MON: off
+------------------+------+------+------+-------+-------+----------+----------
Traffic Class       CBQ-CR CBQ-BR Police Avg Out  Avg In    Packets    Packets
                    (kbps) (kbps) (kbps)  (kbps)  (kbps)      Fwded    Dropped
+------------------+------+------+------+-------+-------+----------+----------
def-in                 -      -   1100   999.7   999.7        481          0
d1                     -      -      -  1901.2  1999.6       1096         57
  d1-def               -      -      -   998.9   998.9        576          0
  d1-web               -      -    900   902.2  1000.6        520         57

>
```

related commands:

configure interface bundle qos

applicable models:

All models.

show qos ethernet

This command displays QoS information for the Ethernet interface.

parameter	definition
interface	
0	Ethernet 0 interface
1	Ethernet 1 interface
class	Optionally, specify a QoS class name

syntax:

ethernet < 1 | 2 > [class]

example:

Router/show/qos> **ethernet 1**

screen display example

```
>  show qos ethernet 1

Interface: Ethernet 1 (Bandwidth = 100000Kbps)

Interface Inbound Configuration & Statistics
-------------------------------------------
Policing: off MON: off
>
```

applicable models:

All models.

show qos historical_stats

This command accesses next-level historical QoS statistics display commands.

syntax:

historical_stats

example:

Router/show/qos> **historical_stats**

next-level commands

show qos historical_stats bundle
show qos historical_stats configuration
show qos historical_stats ethernet

applicable models:

All models.

show qos historical_stats bundle

This command displays historical QoS statistics for the bundle or class.

syntax:

bundle < name >

example:

Router/show/qos/historical_stats> **bundle wan1**

related commands:

show qos historical_stats configuration
show qos historical_stats ethernet

applicable models:

All models.

show qos historical_stats configuration

This command displays historical configuration statistics configuration.

syntax:

configuration

example:

Router/show/qos/historical_stats/bundle wan1> **configuration**

screen display example

```
> show qos historical_stats configuration

Historical Statistics Configuration
-----------------------------------

Sample Interval       : 5 minutes
Uploading             : DISABLED

FTP parameters not yet configured

>
```

related commands:

show qos historical_stats bundle
show qos historical_stats ethernet

applicable models:

All models.

show qos historical_stats ethernet

This command displays historical QoS statistics for the Ethernet interface or class.

parameter	definition
ethernet	
0	Ethernet 0 interface
1	Ethernet 1 interface
class	QoS class name Enter a word.

syntax:

ethernet < 0 | 1 > [class]

example:

Router/show/qos/historical_stats> **ethernet1**

screen display example

```
>  show qos historical_stats ethernet 1

Interface: Ethernet 1 (Bw: 100000 kbps)
CBQ OUT: off  MON OUT: off  MON IN: off

Past 4 hour Statistics:
----------------------
 I      End UpTime   Pkts Rx   Pkts Tx      P a c k e t s  D r o p p e d
       Time  mm:ss     Total     Total      Total Q OverFlo   No Bufs       RED
+-+--------+------+---------+---------+----------+---------+---------+-------
--
 1 09:44:17  00:00         0         0          0         0         0         0
 2 09:39:17  00:00         0         0          0         0         0         0
 3 09:34:17  00:00         0         0          0         0         0         0
 4 09:29:17  00:00         0         0          0         0         0         0
 5 09:24:17  00:00         0         0          0         0         0         0
 6 09:19:17  00:00         0         0          0         0         0         0
 7 09:14:17  00:00         0         0          0         0         0         0
 8 09:09:17  00:00         0         0          0         0         0         0
 9 09:04:17  00:00         0         0          0         0         0         0
10 08:59:17  00:00         0         0          0         0         0         0
11 08:54:17  00:00         0         0          0         0         0         0
12 08:49:17  00:00         0         0          0         0         0         0
13 08:44:17  00:00         0         0          0         0         0         0
14 08:39:17  00:00         0         0          0         0         0         0
15 08:34:17  00:00         0         0          0         0         0         0
16 08:29:17  00:00         0         0          0         0         0         0
17 08:24:17  00:00         0         0          0         0         0         0
18 08:19:17  00:00         0         0          0         0         0         0
19 08:14:17  00:00         0         0          0         0         0         0
20 08:09:17  00:00         0         0          0         0         0         0
21 08:04:17  00:00         0         0          0         0         0         0
22 07:59:17  00:00         0         0          0         0         0         0
23 07:54:17  00:00         0         0          0         0         0         0
...

29 07:24:17  00:00         0         0          0         0         0         0
30 07:19:17  00:00         0         0          0         0         0         0
31 07:14:17  00:00         0         0          0         0         0         0
32 07:09:17  00:00         0         0          0         0         0         0
33 07:04:17  00:00         0         0          0         0         0         0
34 06:59:17  00:00         0         0          0         0         0         0
35 06:54:17  00:00         0         0          0         0         0         0
36 06:49:17  00:00         0         0          0         0         0         0
37 06:44:17  00:00         0         0          0         0         0         0
38 06:39:17  00:00         0         0          0         0         0         0
39 06:34:17  00:00         0         0          0         0         0         0
40 06:29:17  00:00         0         0          0         0         0         0
41 06:24:17  00:00         0         0          0         0         0         0
42 06:19:17  00:00         0         0          0         0         0         0
43 06:14:17  00:00         0         0          0         0         0         0
44 06:09:17  00:00         0         0          0         0         0         0
45 06:04:17  00:00         0         0          0         0         0         0
46 05:59:17  00:00         0         0          0         0         0         0
47 05:54:17  00:00         0         0          0         0         0         0

Current Interval Stats:
----------------------
                           0         0          0         0         0         0

>
```

related commands:

show qos historical_stats bundle
show qos historical_stats configuration

applicable models:

All models.

show reverse_telnet

This command displays information about the reverse telnet and serial port configurations.

syntax:

reverse_telnet

example:

Router/show> **reverse_telnet**

screen display example

```
> show reverse_telnet
REVERSE TELNET CONFIGURATION
----------------------------
Reverse Telnet is Disabled
Reverse Telnet Port     : 2001
Active Connection       : No
Telnet Timeout          : 600 Seconds

SERIAL PORT CONFIGURATION
-------------------------
Baud Rate               : 9600
Parity                  : No Parity
Data Bits               : 8
Stop Bits               : 1
Flow Control            : No Flow Control

>
```

applicable models:

All models.

show running-config

This command shows the running system configuration.

syntax:

show running-config

example:

Router/show> **configuration running**

The following screen capture shows the running system configuration. To continue scrolling through this show, press any key. To go back to the command prompt, type **q**, and press **Return**.

screen display example

```
> show running-config
Please wait... (up to a minute)


module  t1 1
    exit t1
module  t1 2
    exit t1
module  t1 3
    exit t1
module  t1 4
    exit t1
interface  ethernet 0
    ip  address 10.10.1.1 255.255.255.0
    ip  multicast
        mode ospfrip2
        exit multicast
    exit ethernet
interface  ethernet 1
    ip  address 192.168.120.1 255.255.255.0
    ip  multicast
        mode ospfrip2
Press any key to continue (q : quit) :
```

related commands:

show startup-config

applicable models:

All models.

show snmp

This command accesses next-level commands for displaying SNMP configurations and status.

syntax:

snmp

example:

Router/show> **snmp**

next-level commands

show snmp communities
show snmp trap-source
show snmp status
show snmp traps
show snmp trap-host

applicable models:

All models.

show snmp communities

This command displays the current SNMP community names and access privileges.

syntax:

communities

example:

Router/show/snmp> **communities**

related commands:

show snmp status
show snmp traps
show snmp trap-host
configure snmp

applicable models:

All models.

show snmp trap-source

This command displays the source address (IP address) for SNMP traps.

syntax:

trap-source

example:

Router/show/snmp> **trap-source**

applicable models:

All models.

show snmp status

This command displays the total number of SNMP data packets sent to and received from each trap host.

syntax:

status

example:

Router/show/snmp> **status**

The following screen display shows snmp status for a single host.

screen display example

```
 show snmp status
 show snmp status
Sys Name: user
Contact: ,Location:
System Up Time: 0 Days. : 16 Hours. : 55 mins. : 19 secs.
enabled Authentication Failure Trap
0 Total Packets Received
     0 Authentication fails (Unknown communities)
     0 Invalid operation for supplied community
     0 Bad versions
     0 ASN Decoding Errors
     0 Pkts received with size > 1500 bytes
     0 Total Set Requests
     0 Total Get Requests
     0 Total Get Next Requests
     0 Total variables requested to access
     0 Total variables requested to set
     0 Total packets dropped silently
0 Total Packets Sent
     0 Pkts sent with Gen Error
     0 Pkts sent with No Such Name error
     0 Pkts sent with Bad values error
     0 Total Get responses sent
     0 Total Traps sent
     0 Pkts sent with size > 1500 bytes
>
```

related commands:

show snmp communities
show snmp traps
show snmp trap-host
configure snmp

applicable models:

All models.

show snmp traps

This command displays current SNMP trap configuration settings.

syntax:

traps

example:

Router/show/snmp> **traps**

The following screen capture shows SNMP trap configuration settings.

screen display example

```
> show snmp traps
Traps Enable/Disable Configurations:
Config  Group ......
  change : enabled
  save : enabled
Environment Group ......
  temperature : enabled
  power : enabled
Snmp Group ......
  auth_fail : enabled
System Group ......
  shutdown : enabled
  login : enabled
  logoff : enabled
  loginfail : enabled
Ethernet Fail Over Group ......
  success : enabled
  failure : enabled
Frame Relay Group ......
  vcstate : disabled
Sntp Group ......
  sntp : enabled
Ospf Group ......
  ifStateChange : disabled
  virtIfStateChange : disabled
  nbrStateChange : disabled
  virtNbrStateChange : disabled
  ifConfigError : disabled
  virtIfConfigError : disabled
  ifAuthFailure : disabled
  virtIfAuthFailure : disabled
  ifRxBadpacket : disabled
  virtIfRxBadpacket : disabled
  ifTxRetransmit : disabled
  virtIfTxRetransmit : disabled
  originateLsa : disabled
  maxAgeLsa : disabled
BGP4 Group ......
  established : disabled
  backward : disabled
...
>
```

related commands:

show snmp communities
show snmp status
show snmp trap-host
configure snmp

applicable models:

All models.

show snmp trap-host

This command displays the names and IP addresses of configured SNMP trap hosts.

syntax:

trap-host

example:

Router/show/snmp> **trap-host**

related commands:

show snmp communities
show snmp status
show snmp traps
configure snmp

applicable models:

All models.

show sntp

This command displays enabled Simple Network Time Protocol (SNTP) clients.

syntax:

sntp

example:

Router> **show sntp**

applicable models:

All models.

show startup-config

This command shows the contents of a configuration file stored in flash memory.

To continue scrolling through this display, press any key. Or, to go back to the command prompt, type **q**, and press **Return**. See screen display example.

parameter	definition
file name	Name of file to be showed The default is system.cfg.

syntax:

configuration stored [< *file name* >]

example:

Router/show> **configuration stored metro.cfg**

The following screen capture shows a stored system configuration. To continue scrolling through this display, press any key. To go back to the command prompt, type **q**, and press **Return**.

screen display example

```
>  show startup-config
# Router system configuration file (.CFG).
#
# Router assumes no responsibility for product reliability,
# performance, or both if the user modifies the .CFG file.  Full
# responsibility for any modification made to the .CFG file, by
# the user, is assumed by the user.
#
# Version: b040104
# File Created: 04/01/2004-11:55:41



module  t1 1
    exit t1
module  t1 2
    exit t1
module  t1 3
    exit t1
module  t1 4
Press any key to continue (q : quit) :
```

related commands:

show running-config

applicable models:

All models.

show system

This command accesses next-level commands for displaying system-level data such as memory specifications, boot-up diagnostics, flash and DRAM memory usage, and event logging.

syntax:

system

example:

Router/show> **system**

next-level commands

show system configuration
show system diagnostics
show system flash
show system licenses
show system logging
show system memory

applicable models:

All models.

show system configuration

This command displays the NCM and IC hardware configuration.

syntax:

configuration

example:

Router/show/system> **configuration**

The following screen capture shows the NCM and IC hardware configuration.

screen display example

```
> show system configuration
System Configuration:
-------------------------------------------------
Hardware Status:

DRAM quantity:  256MB
DRAM type:      SDRAM
Flash:          32MB
Model Number:   OmniAccess 604
Serial Number:  OmniAccess 60424858
Processor ID:   NEC VR5500
Processor Rev:  16
HW Revision:    A
PCB Revision:   A
EPLD Revision:  09

Internal Level 2 Cache:  NO (0000K)
External Level 2 Cache:  NO (0000K)
Level 3 Cache:           NO (0000K)
-------------------------------------------------
WAN Interface ports -
            T1 - 4 ports available
-------------------------------------------------

Software Status:

Application Image Version:  b040104
FLASH BOOT VERSION:     Not Available
EPROM BOOT VERSION:     T1k050703
Mode:              Routing
-------------------------------------------------
Memory Status:

TOTAL DRAM: 0x10000000 bytes
 status   bytes    blocks   avg block  max block
 ------ --------- -------- ---------- ----------
current
   free  154592368       15   10306157 154446240
  alloc  75091088    54542       1376        -
cumulative
  alloc  121096128   823668        147        -
-------------------------------------------------
Flash Status:/flash1 ---------
Total Memory    Free Memory in flash
32768000               12648448
-------------------------------------------------
System Diagnostics Results:
     DRAM Test:                  PASSED
     Flash Memory Test:          PASSED
     Temperature Test:           PASSED
-------------------------------------------------
Hardware Watchdog Timer Status: disabled
-------------------------------------------------
Hardware Watchdog Timer Status: enabled (disabled if it is FPGA rev less than 5.1)
>
```

related commands:

show system diagnostics
show system memory

applicable models:

All models.

show system diagnostics

This command displays the results of the system diagnostics test that occurs during initial power up.

This test includes information on the DRAM, flash memory, IMC, fans, and temperature. The following screen capture shows a typical system diagnostics display.

syntax:

diagnostics

example:

Router/show/system> **diagnostics**

The following screen capture shows DRAM, flash memory, IMC, fan, and temperature diagnostics.

screen display example

```
> show system diagnostics
System Diagnostics Results:
     DRAM Test:                  PASSED
     Flash Memory Test:          PASSED
     Temperature Test:           PASSED
>
```

related commands:

show system configuration
show system flash
show system memory

applicable models:

All models.

show system flash

This command displays flash memory usage data, in bytes.

syntax:

flash

example:

Router/show/system> **flash**

The following screen capture shows the total and free memory.

screen display example

```
> show system flash
/flash1 ---------
Total Memory    Free Memory in flash
32768000               12648448
>
```

related commands:

show system configuration
show system diagnostics
show system memory

applicable models:

All models.

show system licenses

This command displays feature upgrade licenses.

syntax:

licenses

example:

Router-OmniAccess 604> **show system licenses**

screen display example

```
> show system licenses
Advance IPSec VPN Upgrade License Present.
>
```

applicable models:

OmniAccess 602 and OmniAccess 604

show system logging

This command accesses next-level commands for displaying console logging message prioritization and syslog parameters.

syntax:

logging

example:

Router/show> **system logging**

next-level commands

show system logging commandLog
show system logging console
show system logging syslog

applicable models:

All models.

show system logging commandLog

This command displays all configuration commands entered by the user.

The commands are listed by date and time, starting with the most current. The system can save up to 500 items at a time. See (screen display example).

syntax:

commandLog

example:

Router> **show system logging commandLog**

screen display example

```
> show  system logging commandLog
05/08/04-09:43:53      qos
05/08/04-09:43:39    interface ethernet 0
05/08/04-09:43:36    interface ethernet0
05/08/04-09:43:32    interface ethernet0
05/08/04-08:56:19   save
05/08/04-08:56:16    ip route 172.16.0.0 255.255.255.0 10.20.1.1 5
05/08/04-08:56:07    ip route 172.16.0.0 255.255.255.0 10.10.1.1
05/08/04-08:55:49    ip route 172.16.0.0 255.255.255.0 172.16.0.0
05/08/04-08:55:20    ip route 10.10.1.1 255.255.255.0 172.16.0.0
05/08/04-08:13:11   config t
05/07/04-18:43:56   show flash
>
```

applicable models:

All models.

show system logging console

This command displays console logging message prioritization.

syntax:

logging console

example:

Router/show> **system logging console**

screen display example

```
>  show  system logging console

Logging Console Level: critical

>
```

related commands:

show system logging syslog

applicable models:

All models.

show system logging syslog

This command displays the syslog status, the syslog host IP address, and configured facility priority levels.

syntax:

logging syslog

example:

Router/configure> **system logging syslog**

screen display example

```
>  show system logging syslog

-----------------------------------
Syslog Setting
-----------------------------------
Syslog:                          Enabled
Host IP Address:                 104.154.0.0
Host UDP Port:                   514
Facility Priority Setting:
        facility                 priority
        ========                 ========
        auth:                    warning
        bootp:                   warning
        daemon:                  warning
        domainname:              warning
        gated:                   warning
        kern:                    warning
        mail:                    warning
        ntp:                     warning
        system:                  warning
        fr:                      warning
        ppp:                     warning
        ipmux:                   warning
        bundle:                  warning
        qos:                     warning
        hdlc:                    warning
        local7:                  warning
        firewall:                warning
>
```

related commands:

show system logging console

applicable models:

All models.

show system memory

This command displays DRAM memory usage data, including number of free and allocated bytes, number of fragmented memory blocks, average and maximum block sizes, and a cumulative summary of allocated memory.

syntax:

memory

example:

Router/show/system> **memory**

screen display example

```
> show system memory


TOTAL DRAM: 0x10000000 bytes
 status   bytes    blocks   avg block  max block
 ------ --------- -------- ---------- ----------
current
   free  154592336       16    9662021 154446240
  alloc  75091120    54542       1376        -
cumulative
  alloc  126216944   930011        135        -
>
```

related commands:

show system configuration
show system diagnostics
show system flash

applicable models:

All models.

show tech-support

Saves show and debug information in a file named show_tech.txt.

syntax:

<layer _info>

parameter	definition
layer_info	Displays information for the specified layer. Specify layer1, layer2, layer3 or all.

example:

Router/show/system> **show tech-support**

screen display example

```
>The process will take few minutes....
Press 'y' to continue ? (y/n) : n
router> show tech-support
WARNING:The system information will be stored in file.
                The process will take few minutes....
Press 'y' to continue ? (y/n) : y

>
```

applicable models:

All models.

show telnet

This command shows the configured Telnet timeout in seconds.

syntax:

telnet

example:

Router/show> **telnet**

screen display example

```
>  show telnet
Telnet Server Setting
---------------------
Telnet Server  : Enabled
Telnet Timeout : 900 Seconds
>
```

related commands:

configure telnet_timeout
telnet

applicable models:

All models.

show tftp_server_info

This command shows the TFTP server status.

syntax:

tftp_server_info

example:

Router> **show tftp_server_info**

screen display example

```
Router/configure> show tftp_server_info
             TFTP Setting:
             -------------------------------------
             TFTP server status: Enabled
             Rexmt-interval: 5 seconds
             Max-timeout: 25 seconds
             R1/configure>
Router/configure>
```

applicable models:

All models.

show users

This command displays users logged on to the Router system.

syntax:

users

example:

Router/show> **users**

The following screen capture shows two active users (one via a Telnet connection, one via the console port). The user names, access levels, login dates and times appear, along with the login ports (Router system console port, or IP address of the connecting workstation for a Telnet access).

screen display example

```
> show users


ACTIVE USERS:

SNO     NAME          LOGIN TIME                IPADDR
+-----------------------------------------------------+
 1 : user      :  SAT MAY 08 15:47:16 2004  : CONSOLE
+-----------------------------------------------------+
>
```

related commands:

show user_accounts
configure user

applicable models:

All models.

show user_accounts

This command displays configured user names and access levels.

syntax:

user_accounts

example:

Router/show> **user_accounts**

The following screen capture shows configured user names and access levels.

screen display example

```
> show user_accounts


ALL USERS:

SNO         NAME               LEVEL
+---------------------------------+
 1 : user                   :  1
+---------------------------------+
>
```

related commands:

show users
configure user

applicable models:

All models.

NOTE: You must be logged in as the System Administrator (Level 1 access) to view this data.

show version

This command accesses next-level commands for viewing hardware and software design versions.

syntax:

version

example:

Router/show> **version**

next-level commands

show version IC
show version NCM

applicable models:

All models.

show version IC

This command displays the assembly revision level, PCB revision level, boot image, and software image for a specified interface card (IC). You may need this information when obtaining assistance from customer service.

parameter	definition
IC number	Slot location of the desired module (1 or 2, depending upon the system model)

syntax:

version IC < 1 | 2 >

example:

Router/show> **version IC 1**

related commands:

show version NCM

applicable models:

All models, except OmniAccess 604.

show version NCM

This command displays the assembly revision level, PCB revision level, boot image, and software image for the network controller module (NCM).

Note this information. You may need it when obtaining assistance from customer service.

syntax:

version NCM

example:

Router/show> **version NCM**

screen display example

```
>  show version 0
HW Assembly Version:    A
PCB Revision:           A
FLASH BOOT VERSION:     Not Available
EPROM BOOT VERSION:     T1k050703
OPAL SW VERSION:        b040104
>
```

related commands:

show version IC

applicable models:

All models, except OmniAccess 604.

show virtual-access

Displays information about PPPoE traffic.

syntax:

virtual-access <sessions <id> | <ppp <context | messages-stats | statistics>

example:

Router /show> **virtual-access ppp statistics**

screen display example

```
> show virtual-access ppp statistics
Connection Up Time            : 0 Days 0:04:20

 1. Local MRU :    1500   2. Remote MRU :1500
 3. Local Asyncmap: 0   4. Remote Asyncmap : 0
 5. Local Magic:    41c6  6. Remote Magic : c4d72d4c
 7. Local ProtComp:No   8. Remote ProtComp  :No
 9. Local AddCtrlComp :No 10. Remote AddCtrlComp:No
11. Pkts Received:13  12. Pkts Sent  : 13
13. Bytes Received :17514. Bytes Sent :  222
15. Errs In :0 16. Errs Out :  0
17. Discards In:0  18. Discards Out: 0
19. Unknown Proto :0  20. Bad FCS : 0
21. Bad Addr:0 22. Bad Ctrl: 0
23. Bad High Proto Byte:0  24. Bad Low Proto Byte:  0
25. LongPkts :0  26. MTU : 1500
27. MBufs Alloc :26  28. MBufs Freed: 26
29. buf Alloc Failures:0 30. Traffic Rpt Period:  1
```

related commands:

show interface virtual-access

applicable models:

OmniAccess 601, OmniAccess 602, and OmniAccess 604.

show vlanfwd

This command accesses next-level commands for viewing VLAN management, statistics, and tables.

syntax:

vlanfwd

example:

Router> **show vlanfwd**

next-level commands

show vlandfwd management
show vlanfwd statistics
show vlanfwd table
show vlanfwd tagtable

applicable models:

All models.

show vlanfwd macbridge

This command accesses next-level VLAN bridging display commands.

syntax:

macbridge

example:

Router> **show vlanfwd macbridge**

next-level commands

show vlanfwd macbridge all
show vlanfwd macbridge config
show vlanfwd macbridge dynamic
show vlanfwd macbridge specific
show vlanfwd macbridge static
show vlanfwd macbridge statistics

applicable models:

All models.

show vlanfwd macbridge all

This command shows all MAC entries in the forwarding database.

syntax:

all

example:

Router> **show vlanfwd macbridge all**

related commands:

show vlanfwd macbridge config
show vlanfwd macbridge dynamic
show vlanfwd macbridge specific
show vlanfwd macbridge static
show vlanfwd macbridge statistics

applicable models:

All models.

show vlanfwd macbridge config

This command displays information about all parameters configured for VLAN bridging.

syntax:

config

example:

Router> **show vlanfwd macbridge config**

screen display example

```
>  show vlanfwd macbridge config

Macbridge:                    Disabled
MacAge(in minutes):           5
>
```

related commands:

show vlanfwd macbridge all
show vlanfwd macbridge dynamic
show vlanfwd macbridge specific
show vlanfwd macbridge static
show vlanfwd macbridge statistics

applicable models:

All models.

show vlanfwd macbridge dynamic

This command shows information about all dynamically learned MAC entries in the forwarding database.

syntax:

dynamic

example:

Router> **show vlanfwd macbridge dynamic**

related commands:

show vlanfwd macbridge all
show vlanfwd macbridge config
show vlanfwd macbridge specific
show vlanfwd macbridge static
show vlanfwd macbridge statistics

applicable models:

All models.

show vlanfwd macbridge specific

This command shows information about a specific MAC entry in the forwarding database.

syntax:

specific

example:

Router> **show vlanfwd macbridge specific 00:50:52:02:04:06**

related commands:

show vlanfwd macbridge all
show vlanfwd macbridge config
show vlanfwd macbridge dynamic
show vlanfwd macbridge static
show vlanfwd macbridge statistics

applicable models:

All models.

show vlanfwd macbridge static

This command shows information about all static MAC entries in the forwarding database.

syntax:

static

example:

Router> **show vlanfwd macbridge static**

related commands:

show vlanfwd macbridge all
show vlanfwd macbridge config
show vlanfwd macbridge dynamic
show vlanfwd macbridge specific
show vlanfwd macbridge statistics

applicable models:

All models.

show vlanfwd macbridge statistics

This command shows bridging statistics information for all VLAN interfaces in the system.

syntax:

statistics

example:

Router> **show vlanfwd macbridge statistics**

Definitions

- PktsReceived
  The total number of packets received on a given interface.
- PktsFlooded:
  The number of packets received on a given interface whose destination MAC addresses could not be matched in the forwarding database, and therefore were flooded on all VLAN interfaces.
- PktsForwardeded:
  The number of packets received on a given interface whose destination MAC addresses could be matched in the forwarding database, and therefore were selectively forwarded to a specific VLAN interface.

related commands:

show vlanfwd macbridge all
show vlanfwd macbridge config
show vlanfwd macbridge dynamic
show vlanfwd macbridge specific
show vlanfwd macbridge static

applicable models:

All models.

show vlanfwd management

This command displays the VLAN management settings for all route types.

The management table includes the destination IP address, the gateway MAC address, the expiration time for dynamic VLAN entries, interface type (Ethernet or bundle), and the type of route (static, default, or static). The VLAN ID number is also indicated.

syntax:

management

example:

Router> **show vlanfwd management**

related commands:

show vlanfwd statistics
show vlanfwd table
show vlanfwd tagtable

applicable models:

All models.

show vlanfwd statistics

This command shows if VLAN forwarding is enabled or disabled and the statistics for each VLAN ID.

Statistics are given for incoming, outcoming, and dropped packets.

syntax:

statistics

example:

Router> **show vlanfwd statistics**

related commands:

show vlanfwd management
show vlanfwd table
show vlanfwd tagtable

applicable models:

All models.

show vlanfwd table

This command shows the VLAN table.

syntax:

table

example:

Router> **show vlanfwd table**

related commands:

show vlanfwd management
show vlanfwd statistics
show vlanfwd tagtable

applicable models:

All models.

show vlanfwd tagtable

This command shows the VLAN tagged interface table.

syntax:

tagtable

example:

Router> **show vlanfwd tagtable**

related commands:

show vlanfwd management
show vlanfwd statistics
show vlanfwd table

applicable models:

All models.

show vlanfwd macbridge

This command accesses next-level commands for displaying mac bridging information .

syntax:

macbridge

example:

Router> **show vlanfwd macbridge**

next-level commands

show vlanfwd macbridge all
show vlanfwd macbridge config
show vlanfwd macbridge dynamic
show vlanfwd macbridge specific
show vlanfwd macbridge static
show vlanfwd macbridge statistics

applicable models:

All models.

show vlanfwd macbridge all

This command shows all MAC entries in the forwarding database.

syntax:

all

example:

Router> **show vlanfwd macbridge all**

related commands:

show vlanfwd macbridge config
show vlanfwd macbridge dynamic
show vlanfwd macbridge specific
show vlanfwd macbridge static
show vlanfwd macbridge statistics

applicable models:

All models.

show vlanfwd macbridge config

This command shows configuration information for the MAC bridge.

syntax:

config

example:

Router> **show vlanfwd macbridge config**

related commands:

show vlanfwd macbridge all
show vlanfwd macbridge dynamic
show vlanfwd macbridge specific
show vlanfwd macbridge static
show vlanfwd macbridge statistics

applicable models:

All models.

show vlanfwd macbridge dynamic

This command shows all dynamic MAC entries in the forwarding database.

syntax:

dynamic

example:

Router> **show vlanfwd macbridge dynamic**

related commands:

show vlanfwd macbridge all
show vlanfwd macbridge config
show vlanfwd macbridge specific
show vlanfwd macbridge static
show vlanfwd macbridge statistics

applicable models:

All models.

show vlanfwd macbridge specific

This command shows a specific MAC entry in the forwarding database.

parameter	definition
macaddress	a specific MAC address

syntax:

specific macaddress < macaddress >

example:

Router> **show vlanfwd macbridge specific ### enter a real mac address ###**

related commands:

show vlanfwd macbridge all
show vlanfwd macbridge config
show vlanfwd macbridge dynamic
show vlanfwd macbridge static
show vlanfwd macbridge statistics

applicable models:

All models.

show vlanfwd macbridge static

This command shows all static MAC entries in the forwarding database.

syntax:

static

example:

Router> **show vlanfwd macbridge static**

related commands:

show vlanfwd macbridge all
show vlanfwd macbridge config
show vlanfwd macbridge dynamic
show vlanfwd macbridge specific
show vlanfwd macbridge statistics

applicable models:

All models.

show vldfwd

This command accesses next-level display commands for showing vldfwd information.

syntax:

vldfwd

example:

Router> **show vldfwd**

related commands:

show vldfwd statistics
show vldfwd table
show vldfwd tagtable

applicable models:

All models.

show vldfwd statistics

This command displays the statistics for a specified vld or all configured vlds (default).

parameter	definition
vld_id	Vld identification The range is 1 - 4095; the default is all. Enter a single vld_id <100> or a range of vld_ids < 200-300>.

syntax:

statistics [vld_id < n | n - n >]

example:

Router> **show vldfwd statistics 100**

related commands:

show vldfwd table
show vldfwd tagtable

applicable models:

All models.

show vldfwd table

This command displays the vld table for the specified vld or all configured vlds (default).

parameter	definition
vld_id	Vld identification The range is 1 - 4095; the default is all. Enter a single vld_id <100> or a range of vld_ids < 200-300>.

syntax:

table [vld_id < n | n - n >]

example:

Router> **show vldfwd table 100**

related commands:

show vldfwd statistics
show vldfwd tagtable

applicable models:

All models.

show vldfwd tagtable

This command displays the vld tagged interface table.

syntax:

tagtable

example:

Router> **show vldfwd tagtable**

related commands:

show vldfwd statistics
show vldfwd table

applicable models:

All models.

show vrrp

This command displays VRRP related information.

parameter	definition
group	VRRP group The range is 1 - 255; the default is all groups.
interface	
ethernet0	Ethernet 0 interface
ethernet1	Ethernet 1 interface
mode	
summary	Summary mode
detailed	Detailed mode

syntax:

vrrp [group < n >] [interface < interface >] [mode < mode >]

example:

Router> **show vrrp**

screen display example

```
>  show vrrp
Ethernet 0
    VRRP Mode: 0 (Gratuitous Arp)
Ethernet 1
    VRRP Mode: 0 (Gratuitous Arp)
>
```

applicable models:

All models.

show whoami

This command displays the current user name and access level.

syntax:

whoami

example:

Router/show> **whoami**

screen display example

```
>  show whoami
you are Router, level 1 user
>
```

applicable models:

All models.

7 FILE

Use the **file** commands to copy, delete, and display files located in IC, NCM flash memory (also referred to as system flash or main flash), or compact flash (Model OmniAccess 601 only). Specify compact flash with `/cf0` in the path name. Specify system flash with `/flash1`.

The **file** commands may also be used to upload files to a network host or download files to flash memory.

The first-level **file** commands are as follows:

File Commands

file compare_boot
file copy
file copy_boot
file download
file download_ic
file format
file invalidate_boot
file ls
file rename
file rm
file show_boot
file upload
file version

NOTE: To use the **file** commands, you must log in as a level 1 user. A level 2 user may access the **file ls** command for displaying files currently stored on the IC or NCM modules. The **file** commands are not available to level 3 and level 4 users.

file compare_boot

This command compares bootrom files between the flash and the bootrom image areas.

parameter	definition
srcfile	File name in the flash to be compared.
slot_no	Slot number The range is 0 - 2; the default is 0 for main flash.

syntax:

compare_boot srcfile [slot_no < 0 | 1 | 2 >]

example:

Router> **file compare_boot image.bin 1**

applicable models:

All models.

file copy

This command copies a file in main flash or compact flash memory and assigns the copy a new name. The system will display a warning that compact flash must not be removed during this process and that you must not reboot the router during the copy process. The system will also inform you that copying files requires 3-5 minutes per megabytes of data.

parameter	definition
srcfile	Source file to be copied
destfile	Destination filename

syntax:

copy srcfile < *name* > destfile < *name* >

example:

The following example shows how to copy the file `sys.cfg` to compact flash under the name `oldsys.cfg`.

Router> **file copy sys.cfg /cf0 oldsys.cfg**

applicable models:

All models.

file copy_boot

This command copies the bootrom image to the flash bootrom image area.

parameter	definition
srcfile	File name in flash file system
slot_no	Slot number The range is 0 - 2; the default is 0 for main flash.

syntax:

copy_boot srcfile < name > [slot_no < 0 | 1 | 2 >]

example:

Router> **file copy_boot bimage.bin**

applicable models:

All models.

file download

This command downloads (via TFTP) a file from the network to main (system) flash or compact flash. The host root path is from TFTP root on the host.

parameter	definition
hostIPA	Host IP address
scrfile	Name of the source file
dstfile	Name of the destination file

syntax:

download hostIPA < ip address > srcfile < name > dstfile < name >

example:

Router> **file download 10.10.22.3 image.Z image2.Z**

applicable models:

All models.

file download_ic

This command downloads a file from main flash to IC flash.

parameter	definition
slot_no	Slot number The range is 0 - 2; the default is 0 for main flash.
srcfile	Name of the source file
destfile	Name of the destination file

syntax:

download_ic < slot_no 0 | 1 | 2 > srcfile < name > dstfile < name >

example:

Router> **file download 1 image.Z image2.Z**

applicable models:

All models except OmniAccess 604.

file format

This command formats main flash or compact flash.

parameter	definition
slot_no	Slot number The range is 0 - 2; the default is 0 for main flash.

syntax:

format [slot_no < 0 | 1 | 2 >]

example:

Router> **file format 2**

applicable models:

All models.

file invalidate_boot

This file invalidates the flash bootrom inage.

parameter	definition
slot_no	Slot number The range is 0 - 2; the default is 0 for main flash.

syntax:

invalidate_boot [slot_no < 0 | 1 | 2 >]

example:

Router> **file invalidate_boot 2**

applicable models:

All models.

file ls

This command lists the files in main flash or compact flash.

parameter	definition
slot_no	Slot number The range is 0 - 2; the default is 0 for main flash. compact flash is cf0.

syntax:

ls [slot_no < 0 | 1 | 2 >]

example:

host> **file ls**

host> **file ls /cf0**

host/file> **ls /cf0**

```
CONTENTS OF /cf0:

  size          date       time       name
--------       ------     ------    --------
    1813    JAN-12-2005  15:18:00   system.cfg

Total bytes:     1813
Bytes Free on /cf0:  63807488
```

applicable models:

All models.

For compact flash commands, all models with compact flash slots.

file rename

This command changes the name of a file in main flash or compact flash.

syntax:

rename <old name> <new name>]

parameter	definition
old name	The name of the file to be renamed.
new name	The new name of the flash file.

example:

Router> **file rename main main_old**

applicable models:

All models.

file rm

This command removes a file from main flash or compact flash.

syntax:

rm file_name [slot_no < 0 | 1 | 2 >]

parameter	definition
file_name	The name of the file to be deleted.
slot_no	Slot number The range is 0 - 2; the default is 0 for main flash.

example:

Router> **file rm conf.bin 1**

applicable models:

All models.

file show_boot

This command displays flash bootrom data.

parameter	definition
slot_no	Slot number The range is 0 - 2; the default is 0 for main flash.

syntax:

show_boot [slot_no < 0 | 1 | 2 >]

example:

Router> **file show_boot**

applicable models:

All models.

file upload

This command uploads files from a Router system to a network host.

These commands use the TFTP protocol.

syntax:

upload

example:

Router> **file upload**

next-level commands

file upload event_log
file upload flash_file

applicable models:

All models.

file upload event_log

This command uploads the Router system event log to a network host.

parameter	definition
hostname	Host IP address where file is to be uploaded
filename	Name of the remote file to upload

syntax:

event_log hostname < *IP address* > filename < *name* >

example:

Router> **file upload event_file 10.1.100.22 events**

related commands:

file upload flash_file

applicable models:

All models.

file upload flash_file

This command uploads a flash memory file to a network host.

TABLE 7

parameter	definition
hostname	Destination host IP address
srcfile	Name of the uploaded file.
destfile	Name of the destination file on the server and its path

syntax:

flash_file hostname < *IP address* > srcfile < *name* > destfile < *name* >

example:

Router> **file flash_file 10.2.200.1 system.cfg config01.cfg**

related commands:

file upload event_log

applicable models:

All models.

file version

This command displays the versions of files in main flash or compact flash.

syntax:

version

example:

Router> **file version**

related commands:

display version

applicable models:

All models.

8 EXIT

Use the **exit** command to exit the system CLI or to exit from a lower-level command prompt to a higher-level command prompt in the CLI hierarchy.

The **exit** command is as follows:

Exit Command

exit

exit

This command exits the system CLI, or exits from a lower-level command prompt to a higher-level command prompt in the CLI hierarchy.

parameter	definition
levels	Number of levels to exit. The range is 1 - 6; the default is 1.

syntax:

exit [levels < n >]

example:

Router-T1/configure/interface/bundle wan1> **exit 2**

Router-T1/configure>

In the example above, the **exit** command is used to exit two levels from the configure/interface/ bundle> prompt to the configure> prompt.

If you enter **exit** at the Router-T1> prompt, press either **y** to exit the system CLI or **n** to return to the Router-T1> prompt.

applicable models:

All models.

shutdown

Shuts down the router and terminates all processes. User connections to the router are terminated.

syntax:

shutdown

example:

Router> **shutdown**

applicable models:

All models.

9
FILTER LIST

configure ip access-list add

Adds a new rule to an IP packet filtering rule set. Each rule is added with an identifying line number for future editing or deletion.

parameter	definition
rul_action	
permit	Allows access if conditions are matched.
deny	Discards access if conditions are matched.
reject	Discards a packet, and sends an ICMP host unreachable message.
protocol	Name or number of an Internet protocol. This can be one of the key words (TCP, UDP, ICMP, or IP), or an integer int he range 0 - 255 representing an IP protocol number. To match any Internet protocol, including ICMP, TCP, or UDP, use the keyword IP.
source	Source host or network address. Or, type **any** to specify a source address/wildcard of 0.0.0.0/255.255.255.255 or 0.0.0.0/32.
/wildcard	Optional wildcard bits to be applied to the source address or destination address. This entry can be an IP address (as in Cisco notation) or the number of bits (as in NetBlazer notation).
destination	Destination host or network address. Or, type **any** to specify a destination address/wildcard of 0.0.0.0/255.255.255.255 or 0.0.0.0/32.
sport	Optional entry for TCP and UDP protocols; allows the source port to be used for packet filtering. Use comparison symbols to specify ports as follows:
=p	Port number **p**, where n is 0 - 65535.
!=p	Excludes port **p**.
>p	Any port number greater than **p**
>=p	Any port number greater than or equal to **p**
<p	Any port number less than **p**
<=p	Any port number less than or equal to **p**
p1-p2	Any port number within the range **p1** - **p2**
dport	Optional entry for TCP and UDP protocols; allows the destination port to be filtered. Use the same comparison symbols and port numbering described above for the source port (sport).
precedence	Optional numeric entry; allows IP header precedence field to be filtered. The range is 0 - 7.
flags	Allows TCP flags to be filtered (optional).You can specify multiple TCP flags by separating the above keywords with commas (no spaces allowed). This entry may be any of the following words:
established	Used to match an established connection (Cisco-compatible).
fin	Matches the TCP FIN header flag.
syn	Matches the TCP SYN header flag.
ack	Matches the TCP ACK header flag.
psh	Matches the TCP PSH header flag.

rst	Matches the TCP RST header flag.
urg	Matches the TCP URG header flag.
icmptype	Optional numeric entry for ICMP protocol, allowing the ICMP message type to be filtered (optional, range is 0 - 255).
icmpcode	Optional numeric entry for ICMP protocol, allowing the ICMP message code to be filtered, if specified along with a message type. The range is 0 - 255.
tos	Type of service (optional numeric entry for UDP and ICMP protocols). Allows the IP header TOS field to be filtered. The range is 0 - 15.
log	Allows a logging message to be reported to the user when a rule match occurs (optional).
on	Log the matching packet on.
off	Log the matching packet off (default).
expire	The amount of time (in seconds) before a rule expires.
finterface	The name of the forwarding interface (Ethernet/bundle).
faddr	Forwarding interface IP address.
rule_type	The filter rule type (for example, MHU daemon added filter rule).

syntax:

Syntax for TCP

[no] add rul_action < permit | deny > protocol < tcp > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [sport < 0 - 65535 >] [dport < 0 - 65535 >] [precedence < 0 - 7 >] [tos < 0 - 15 >] [flags < established | fin | syn | ack | psh | rst | urg >] [log < on | off >]

Syntax for UDP

[no] add rul_action < permit | deny > protocol < udp > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [sport < 0 - 65535 >] [dport < 0 - 65535 >] [precedence < 0 - 7 >] [tos < 0 - 15 >] [log < on | off >]

Syntax for ICMP

[no] add rul_action < permit | deny | reject > protocol < icmp > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [icmptype < 0 - 255 >] [icmpcode < 0 - 255 >] [precedence < 0 - 7 >] [tos < 0 - 15 >] [log < on | off >]

Syntax for IP

[no] add rul_action < permit | deny > protocol < ip > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [precedence < 0 - 7 >] [tos < 0 - 15 >] [log < on | off >]

example:

a Router/configure/ip/filter_list Rules_01> **add permit tcp 10.1.3.9/12 10.1.100.46/255.255.255.0 dport =23 flags established log on**

b Router/configure/ip/filter_list Rules_02> **add deny icmp 10.1.20.9/32 any icmptype 8 icmpcode 0 log on**

c Router/configure/ip/filter_list Rules_03> **add permit udp 10.1.10.7 any sport =12 precedence 3 tos 2**

d Router/configure/ip/filter_list Rules_04> **add deny ip 10.1.100.2 255.255.255.0 precedence 5 tos 1 log on**

The above command entries add four filtering rules to a filtering rule set (one each for the TCP, UDP, ICMP, and IP protocols).

related commands:

configure ip access-list delete
configure ip access-list insert

applicable models:

All systems.

configure ip access-list delete

This command deletes a rule from an IP packet filtering rule set.

You must know the line number of the rule to delete it. To display the rules currently stored in a rule set before or after deleting them, use the **show ip access-list** command. When you delete a rule, the line numbers of successive rules in the list are revised accordingly.

parameter	definition
rul_lineno	The line number of a specific rule set The range is 1 - 65535.

syntax:

delete rul_lineno < n >

example:

Router/configure/ip/access-list Rules_01> **delete 3**

related commands:

configure ip access-list add
configure ip access-list insert

applicable models:

All systems.

configure ip access-list insert

This command inserts a new rule into an existing filtering rule set. This command lets you enter a rule in the middle of an existing rules set by specifying a line number.

You can display the contents of the current filtering rule set using the **show ip access-list filter_list** command. Each rule has a unique line number for identification purposes.

parameter	definition
rule_lineno	Line number The range is 1 - 65535.
rul_action	
permit	Allows access if conditions are matched.
deny	Discards access if conditions are matched.
reject	Discards a packet, and sends an ICMP host unreachable message.
protocol	Name or number of an Internet protocol. This can be one of the key words (TCP, UDP, ICMP, or IP), or an integer int he range 0 - 255 representing an IP protocol number. To match any Internet protocol, including ICMP, TCP, or UDP, use the keyword IP.
source	Source host or network address. Or, type **any** to specify a source address/wildcard of 0.0.0.0/255.255.255.255 or 0.0.0.0/32.
/wildcard	Optional wildcard bits to be applied to the source address or destination address. This entry can be an IP address (as in Cisco notation) or the number of bits (as in NetBlazer notation).
destination	Destination host or network address. Or, type **any** to specify a destination address/wildcard of 0.0.0.0/255.255.255.255 or 0.0.0.0/32.
sport	Optional entry for TCP and UDP protocols; allows the source port to be used for packet filtering. Use comparison symbols to specify ports as follows:
=p	Port number **p**, where n is 0 - 65535.
!=p	Excludes port **p**.
>p	Any port number greater than **p**
>=p	Any port number greater than or equal to **p**
<p	Any port number less than **p**
<=p	Any port number less than or equal to **p**
p1-p2	Any port number within the range **p1** - **p2**
dport	Optional entry for TCP and UDP protocols; allows the destination port to be filtered. Use the same comparison symbols and port numbering described above for the source port (sport).
precedence	Optional numeric entry; allows IP header precedence field to be filtered. The range is 0 - 7.
flags	Allows TCP flags to be filtered (optional).You can specify multiple TCP flags by separating the above keywords with commas (no spaces allowed). This entry may be any of the following words:
established	Used to match an established connection (Cisco-compatible).
fin	Matches the TCP FIN header flag.
syn	Matches the TCP SYN header flag.
ack	Matches the TCP ACK header flag.
psh	Matches the TCP PSH header flag.
rst	Matches the TCP RST header flag.

urg	Matches the TCP URG header flag.
icmptype	Optional numeric entry for ICMP protocol, allowing the ICMP message type to be filtered (optional, range is 0 - 255).
icmpcode	Optional numeric entry for ICMP protocol, allowing the ICMP message code to be filtered, if specified along with a message type. The range is 0 - 255.
tos	Type of service (optional numeric entry for UDP and ICMP protocols). Allows the IP header TOS field to be filtered. The range is 0 - 15.
log	Allows a logging message to be reported to the user when a rule match occurs (optional).
on	Log the matching packet on.
off	Log the matching packet off (default)

syntax:

Syntax for TCP

[no] insert rule_lineno < 1 - 65535 > rul_action < permit | deny > protocol < tcp > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [sport < 0 - 65535 >] [dport < 0 - 65535 >] [precedence < 0 - 7 >] [tos < 0 - 15 >] [flags < established | fin | syn | ack | psh | rst | urg >] [log < on | off >]

Syntax for UDP

[no] insert rule_lineno < 1 - 65535 > rul_action < permit | deny > protocol < udp > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [sport < 0 - 65535 >] [dport < 0 - 65535 >] [precedence < 0 - 7 >] [tos < 0 - 15 >] [log < on | off >]

Syntax for ICMP

[no] insert rule_lineno < 1 - 65535 > rul_action < permit | deny | reject > protocol < icmp > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [icmptype < 0 - 255 >] [icmpcode < 0 - 255 >] [precedence < 0 - 7 >] [tos < 0 - 15 >] [log < on | off >]

Syntax for IP

[no] insert rule_lineno < 1 - 65535 > rul_action < permit | deny > protocol < ip > source < *IP address* > [/ < *wildcard* >] | any destination < *IP address* > [/ < *wildcard* >] | any [precedence < 0 - 7 >] [tos < 0 - 15 >] [log < on | off >]

example:

Router/configure/ip/filter_list Rules_01> **insert 4 permit tcp 10.1.10.7 any dport 50-55 precedence 4 tos 6**

The example above inserts a new rule, 4, behind existing rule 3 in the rule set. It also increments the line numbers of all successive rules in that set.

related commands:

configure ip access-list add
configure ip access-list delete

applicable models:

All systems.

10

GENERIC ROUTING ENCAPSULATION COMMANDS

Use the Generic Routing Encapsulation (GRE) commands to configure an tunnel interface on a Router system.

interface tunnel

Names a tunnel interface.

syntax:

tunnel_name

parameter	definition
tunnel_name	Names can be up to eight characters in length

example:

To create the tunnel named **mainpipe**, enter:

```
Router/configure>  interface tunnel mainpipe
configuring new tunnel interface
 mainpipe=======>
Router/configure/interface/tunnel mainpipe>
```

applicable models:

All models.

interface tunnel ip

Configures the IP address and subnet mask for the interface.

syntax:

[no] ip <ip-address> <subnet mask>

parameter	definition
ip-address	The IP address of the tunnel interface.
subnet mask	The subnet mask of the tunnel interface.

example:

To assign an IP address and subnet mask to the tunnel **mainpipe**, enter:

```
Router/configure/interface/tunnel mainpipe>  ip address
192.168.122.10 255.255.255.0
```

applicable models:

All models.

interface tunnel ip unnumbered

Configure the tunnel interface without an IP address.

syntax:

[no] ip unnumbered <interface_name>

parameter	definition
interface_name	The name of the interface, for example, ethernet0.

example:

The following example shows how to configure the tunnel for the ethernet1 interface without an IP address.

```
Router/configure/interface/tunnel mainpipe> ip unnumbered ethernet1
Router/configure/interface/tunnel mainpipe>
```

NOTE: In the above example, ethernet1 must be configured with an IP address.

applicable models:

All models.

interface tunnel tunnel destination

Configures the IP address for the tunnel destination. The tunnel destination cannot be a point-to-point interface peer. It should be reachable through a physical interface.

syntax:

[no] tunnel destination ip_address

parameter	definition
ip_address	The IP address for the end of the tunnel.

example:

To set the address of the other end of the tunnel **mainpipe** to be 10.10.20.10, enter:

```
Router/configure/interface/tunnel mainpipe>  tunnel destination
10.10.20.10
Router/configure/interface/tunnel mainpipe>
```

applicable models:

All models.

NOTE: The tunnel destination gateway cannot be a virtual access interface.

interface tunnel tunnel source

Configures the IP address for the tunnel source. The tunnel source address is the IP address of a local interface.

syntax:

[no] tunnel source ip_address

parameter	definition
ip_address	The IP address for the end of the tunnel.

example:

To set the source IP address of the tunnel mainpipe to be 10.10.40.10, enter:

Router/configure/interface/tunnel mainpipe> **tunnel source 10.10.40.10**
Router/configure/interface/tunnel mainpipe>

applicable models:

All models.

interface tunnel keepalive

Configures keepalive signals between the tunnel end points. Send a keeplalive every <interval> seconds. If there is no response after <retries> bring the tunnel down.

syntax:

keepalive [interval] [retries]

parameter	definition
interval	The period of time between keepalive signals. Valid range is 0—120 seconds. The default is 10 seconds. Specify 0 seconds for no keepalives.
retries	The number of signals sent without a reply. Valid range is 1—16 tries, the default is 3 tries.

example:

To configure a keepalive signal with 10 second intervals, and retries set to 3, enter:

Router/configure/interface/tunnel mainpipe> **keepalive interval 10 retries 3**
Router/configure/interface/tunnel mainpipe>

applicable models:

All models.

interface tunnel tunnel checksum

Configures end-to-end checksum tests.The default mode is off. Enabling checksums will force the Router router to drop corrupted packets. To disable checksum use **no tunnel checksum.**

syntax:

[no] tunnel checksum

example:

To enable checksum tests on all traffic through the tunnel, enter:

```
Router/configure/interface/tunnel mainpipe>  tunnel checksum
Router/configure/interface/tunnel mainpipe>
```

applicable models:

All models.

interface tunnel tunnel key

Configures an ID key for a tunnel interface. This key must be the same value on both tunnel endpoints. The key field is intended to be used for identifying an individual traffic flow within a tunnel. Tunnel ID keys can be used as a form of weak security to prevent misconfiguration or injection of packets from a foreign source. However this is not a reliable security option. The default mode is no key configured. To disable the key enter **no tunnel key**

syntax:

[no] tunnel key <key_number>

parameter	definition
key_number	The ID key as a 32-bit value. Valid range is 0— 4294967295.

example:

To configure the tunnel key for **mainpipe** to be **884452**, enter:

Router/configure/interface/tunnel mainpipe> **tunnel key 884452**
Router/configure/interface/tunnel mainpipe>

applicable models:

All models.

interface tunnel tunnel mode

Configure the tunnel encapsulation mode.

syntax:

[no] tunnel mode mode-type

parameter	definition
mode-type	The encapsulation method. Can be either GRE (the default) or IPIP (IP over IP).

example:

To configure GRE encapsulation, enter:

```
Router/configure/interface/tunnel mainpipe>  tunnel mode gre
Router/configure/interface/tunnel mainpipe>
```

applicable models:

All models.

interface tunnel tunnel protection

Associates a tunnel with an IPSec profile. IPSec will derive the IPSec peer and proxy information from the **tunnel source** and **tunnel destination** configuration. (Therefore, the tunnel must be configured before applying this command.) This simplifies the configuration since the IPSec peer and the match addresses are no longer needed.

To disable use **no tunnel protection** <policy*_name* > <*key_value*>

syntax:

[no] tunnel protection policy_name key_string

parameter	definition
policy_name	The name of the IPSec policy. Enter a word of no more than eight characters.
key_value	The IPSec policy key value. Enter a string of no more than 49 characters.

To protect the **mainpipe** tunnel with an IPSec policy named **policy28** having a key of **1234567**, enter:

Router/configure/interface/tunnel mainpipe> **tunnel protection policy28 1234567**
Router/configure/interface/tunnel mainpipe>

applicable models:

All models.

interface tunnel tunnel sequence

Configures a tunnel interface to drop out-of-order datagrams. This command is turned off by default. Once enabled, use **no tunnel sequence** to disable.

This command is included to provide compatibility with Cisco Systems.

syntax:

[no] tunnel sequence

parameter	definition
tunnel_name	Names can be up to eight characters in length

example:

To enable the tunnel to delete packets arriving out of order, enter:

```
Router/configure/interface/tunnel mainpipe>  tunnel sequence
Router/configure/interface/tunnel mainpipe>
```

applicable models:

All models.

interface tunnel tunnel ttl

Configures the time-to-live (TTL) value for the tunnel interface. To disable, use **no tunnel ttl** command.

syntax:

[no] tunnel <ttl value>

parameter	definition
ttl_value	The time to live value. Valid range is 1—255 seconds. The default is 30.

example:

To set the TTL value to 45 seconds for the tunnel **mainpipe** enter:

Router/configure/interface/tunnel mainpipe> **tunnel ttl 45**
Router/configure/interface/tunnel mainpipe>

applicable models:

All models.

interface tunnel tunnel tos

Configures the Type of Service (ToS) value for the tunnel interface. If the ToS value is not specified, the ToS value of the inner IP header is copied to the GRE-IP header. Use the **no** form to disable this command.

syntax:

[no] tunnel tos <tos value>

parameter	definition
tos value	The ToS for the tunnel. Valid range is 0-255. The default is 0 (no ToS).

example:

To set the ToS value for the **mainpipe** tunnel to 100, enter:

Router/configure/interface/tunnel mainpipe> **tunnel tos 100**
Router/configure/interface/tunnel mainpipe>

applicable models:

All models.

interface tunnel crypto

Configures the interface for trusted or untrusted network types.

syntax:

[no] crypto network_type

parameter	definition
network_type	The type of network, either trusted or untrusted.

example:

To enable the **mainpipe** tunnel interface for, trusted networks, enter:

Router/configure/interface/tunnel mainpipe> **crypto trusted**
Router/configure/interface/tunnel mainpipe>

applicable models:

All models.

interface tunnel icmp

Configures the behavior of Internet Control Message Protocol over the tunnel.

syntax:

[no] icmp [redirect | unreachable]

parameter	definition
redirect	Controls ICMP traffic redirection. The default is enabled.
unreachable	Controls ICMP traffic processing for unreachable destinations.

example:

To allow ICMP traffic to be redirected over the mainpipe tunnel, enter:

```
Router/configure/interface/tunnel mainpipe> icmp redirect
Router/configure/interface/tunnel mainpipe>
```

To prevent ICMP packets from being forwarded to unreachable destinations, enter:

```
Router/configure/interface/tunnel mainpipe> no icmp unreachable
Router/configure/interface/tunnel mainpipe>
```

applicable models:

All models.

interface tunnel shutdown

Takes down the tunnel interface. To prevent the interface from being shutdown, configure the tunnel with the **no shutdown** command. The default is no **shutdown**.

syntax:

[no] shutdown

example:

To configure the **mainpipe** tunnel interface to be available or to restart it after stopping it, enter:

Router/configure/interface/tunnel mainpipe> **no shutdown**

To stop this interface, enter:

Router/configure/interface/tunnel mainpipe> **shutdown**

Remember that IP routes must be configured before the tunnel can pass traffic.

applicable models:

All models.

show ip interfaces

Displays the statics for the specified interfaces.

syntax:

interfaces [interface]

parameter	definition
interface	Specifies which interface statistics to display. The default displays all interfaces.

example:

To display the configuration for all interfaces, enter:

Router/configure/interface/tunnel mainpipe> **show ip interfaces**

ethernet0 (unit number 0)
Type: ETHERNET_CSMACD
Flags: (0x807c203) UP, MULTICAST-ROUTE
Internet Address: 67.121.71.220
Internet Netmask: 255.255.255.0
Internet Broadcast: 67.121.71.255
Maximum Transfer Unit: 1500 bytes
Mac Address: 00:00:23:00:60:00

null (unit number 2)
Type: NULL
Flags: (0x4041) UP, RUNNING, MULTICAST-ROUTE
ICMP unreachables will not be sent
Mac Address: 00:00:23:00:60:02

mainpipe (unit number 3)
Type: TUNNEL
Flags: (0x44203) UP, MULTICAST-ROUTE
Internet Address: 10.10.10.200
Internet Netmask: 255.255.0.0
Internet Broadcast: 10.10.255.255
ICMP redirects will not be sent
ICMP unreachables will not be sent
Maximum Transfer Unit: 1472 bytes
Keepalive: enabled interval: 10 retries: 3 status: up
Source Address: 10.10.30.10 (no source interface)
Destination Address: 192.169.125.10
Gateway: not available
Protocol: IPIP
Mac Address: 00:00:23:00:60:03
Router/configure/interface/tunnel mainpipe>

To display the statistics for just the tunnel mainpipe, enter:

Router/configure/interface/tunnel mainpipe> **show ip interfaces mainpipe**

```
mainpipe (unit number 3)
Type: TUNNEL
Flags: (0x44203) UP, MULTICAST-ROUTE
Internet Address: 10.10.10.200
Internet Netmask: 255.255.0.0
Internet Broadcast: 10.10.255.255
ICMP redirects will not be sent
ICMP unreachables will not be sent
Maximum Transfer Unit: 1472 bytes
Keepalive: enabled interval: 10 retries: 3 status: up
Source Address: 10.10.30.10 (no source interface)
Destination Address: 192.169.125.10
Gateway: not available
Protocol: IPIP
TTL: 45
TOS: 100
Key Value: 884452
Checksum: enabled
Sequence Datagrams: enabled
Path MTU Discovery: enabled
Mac Address: 00:00:23:00:60:03
Router/configure/interface/tunnel mainpipe>
```

applicable models:

All models.

show interface tunnels

Displays the status and traffic statistics for the specified tunnel.

syntax:

tunnel tunnel_name

parameter	definition
tunnel_name	Specifies which tunnel statistics to display.

example:

To see details on all tunnels, enter:

Router/configure/interface/tunnel mainpipe> **show interface tunnel mainpipe**

```
Tunnel: mainpipe                Status: down (No source interface)
Internet Address: 10.10.10.200      Internet Netmask: 255.255.0.0
Source Address: 10.10.30.10         Destination Address: 192.169.125.10
MTU: 1476 bytes                  Protocol: IPIP
ICMP unreachable: will not be sent ICMP redirect: will be sent
Crypto type: not set             IPSEC/IKE: policy not set key not set
TTL: 45                           Keepalive: enabled, interval:   10, retries
:   3
TOS: 100                      Path MTU discovery: enabled
Key Value: 884452               Checksum: enabled
Sequence Datagrams: enabled

Tunnel Statistics:
   Bytes Rx              0   Bytes Tx              0
   Packets Rx            0   Packets Tx            0
   Err Packets Rx        0   Output Errs           0
```

Router/configure/interface/tunnel mainpipe>

applicable models:

All models.

clear counters tunnel

Resets the tunnel counters to null. To clear all tunnel counters, use **clear counters tunnels**.

syntax:

tunnel tunnel_name

parameter	definition
tunnel_name	Specifies which tunnel counters to reset.

example:

To reset the counters for mainpipe, enter:

Router> **clear counters tunnel mainpipe**

To reset the counters for all tunnels, enter:

Router> **clear counters tunnels**

applicable models:

All models.

debug ip tunnel

Controls the tunnel debug commands.

syntax:

tunnel [message -type]

parameter	definition
message-type	Specifies the type of message to debug. Valid choices are: all debugs all messages decap debugs decapsulation related messages encap debugs encapsulation related messages error debugs all error messages keepalive debugs keepalive messages packet debugs packet trace messages state debugs interface state change messages

example:

To set a debug keepalive process on all tunnels, enter:

Router> **debug ip tunnel keepalive**
Router>

To debug all traffic on all tunnels, enter:

Router> **debug ip tunnel all**
Router>

applicable models:

All models.

11
MULTICAST COMMANDS

This chapter explains how to use the multicast commands in the 8.0 CLI. Use the multicast commands to configure a Router system for mtrace, Internet Group Management Protocol (IGMP), and sparse mode Protocol Independent Multicast (PIM).

ip igmp

Enables IGMP (Internet Group Management Protocol) on the associated interface with the default set of configurations. Use the **no** form of this command to disable IGMP on the interface. IGMP is not enabled by default.

syntax:
[no] ip igmp

Operating Mode:
Interface

example:
To enable IGMP for interface ethernet0, enter:
`Router/configure/ip/igmp> interface ethernet0`

applicable models:

All models.

ip igmp ignore-v1-messages

Controls processing of all IGMPv1 messages on the associated interface. Note that this breaks interoperability with older IGMPv1 users on the network and should be employed when it is important to maintain small group leave latencies. This command is only effective for IGMPv3 and IGMPv2. This command is disabled by default.

syntax:
[no] ip igmp ignore-v1-messages

example:
In the following example, *no ip igmp ignore-v1-messages* is used to enable processing of IGMPv1 messages on interface ethernet1.
Router/configure/ip/igmp/interface ethernet1> **no ignore-v1-messages**

applicable models:

All models.

ip igmp ignore-v2-messages

Controls the processing of all IGMPv2 messages on the associated interface. Note that this breaks interoperability with older IGMPv2 users on the network and should be done when it is important to maintain small group leave latencies. (This command is only effective for IGMPv3.) Use the no form of this command to allow IGMPv2 messages on the specified interface. This command is disabled by default.

syntax:
[no] ip igmp ignore-v2-messages

example:
In the following example, *ip igmp ignore-v2-messages* is used to disable processing of IGMPv2 messages on interface ethernet1.
Router/configure/ip/igmp/interface ethernet1> **ignore-v2-messages**
Router/configure/ip/igmp/interface ethernet1>

applicable models:

All models.

ip igmp last-member-query-count

Configures the Last Member Query Count per RFC 3376. It specifies the number of queries sent out on startup, separated by the Last Member Query Interval. Use the *no* form of the configuration to return to the default *lastmember-query-count* value.
Note: Specifying a *value* in the *no* form has no effect on the configuration.

syntax:
[no] ip igmp last-member-query-count value

parameter	definition
value	An integer between 1 and 10.

example:
The following example configures the Last Member Query Count to be 4.
Router/configure/ip/igmp/interface ethernet1> **last-member-query-count 4**

applicable models:
All models.

ip igmp last-member-query-interval

Specifies the maximum amount of time that hosts have to respond to Group-Specific query messages. The *no* form returns to the default value of the Last Member Query Interval. Note: Specifying a value for *time* in the *no* form has no effect on the configuration.The default is 1000 seconds.

syntax:
[no] ip igmp last-member-query-interval time

parameter	definition
time	a number of seconds between 1 and 3174

example:
Example 1
The following example configures the default Last Member Query Interval to be 30 seconds:
Router/configure/ip/igmp/interface ethernet0> **last-member-query-interval 30**
Router/configure/ip/igmp/interface ethernet0>

applicable models:
All models.

ip igmp query-interval

Specifies the value of the query interval in seconds. The Query Interval is the interval between General Queries sent by the Querier. It is encoded in the Query Interval Code (QQIC) field of General Queries. The Query Interval cannot be less than the Query Response Interval (configured using *ip igmp query-response-interval*).
Use the *no* form of the configuration to return to the default value of 125 seconds. Note: Specifying a value for *time-seconds* in the *no* form has no effect on the configuration.

syntax:
[no] ip igmp query-interval time-seconds?

parameter	definition
time-seconds	a number in seconds between 0 and 3174 Note: This value cannot be less than the Query Response Interval

example:
The following example configures the default Query Interval for the ethernet0 interface to be 60 seconds.
Router/configure/ip/igmp/interface ethernet0> **query-interval 60**
Router/configure/ip/igmp/interface ethernet0>

applicable models:
All models.

ip igmp query-response-interval

*C*onfigures the maximum amount of time before a Host sends a Report message in response to a received General Query. This interval is encoded in the Max Resp Code field of General Query messages.
Use the *no* form of the configuration to return to the default value of 10 seconds. Note: Specifying a value for *time-seconds* in the *no* form has no effect on the configuration.

syntax:
[no] ip igmp query-response-interval time-seconds

parameter	definition
time-seconds	1 and 3174 seconds Note: This value must be less than or equal to the Query Interval.

example:
The following example configures the default Query Response Interval to be 300 seconds (or 3000 deciseconds).
Router/configure/ip/igmp/interface ethernet0> **query-response-interval 300**
Router/configure/ip/igmp/interface ethernet0>

applicable models:
All models.

ip igmp require-router-alert

*S*pecifies whether to ignore messages that do not contain the Router Alert option, thereby improving protocol security. When configured, the following messages are ignored if they do not contain the Router Alert option:
• State-Change Report
• Current-State Report
• Leave Message in IGMP version 2 mode
• Report Message in IGMP version 2 mode
Use the *no* form configuration to return to the default value. This command is disabled by default.

syntax:
[no] ip igmp require-router-alert

example:
The following example turns *require-router-alert* on for interface ethernet0 and off for all other interfaces.
Router/configure/ip/igmp/interface ethernet0> **require-router-alert**
Router/configure/ip/igmp/interface ethernet0>

applicable models:

All models.

ip igmp robustness

Allows for tuning of the IGMP protocol or interface to accommodate a lossy subnet. IGMP is robust to (Robustness - 1) packet losses, and this value is advertised in the Querier's Robustness Variable (QRV) field of Query.
Use the *no* form of this configuration to return to the default value of Robustness (2). Note: Specifying a *value* in the *no* form has no effect on the configuration.
[no] ip igmp robustness value

parameter	definition
value	an integer between 2 and 10.

example:
The following example configures the default Robustness to be 7.
Router/configure/ip/igmp/interface ethernet0> **robustness** 7
Router/configure/ip/igmp/interface ethernet0>

applicable models:
All models.

ip igmp send-router-alert

Specifies whether sent IGMP packets will include the Router Alert option in the IP packet header. This command is enabled by default.

syntax:
[no] ip igmp send-router-alert

example:
The following example turns the send-router-alert option on for ethernet1.
Router/configure/ip/igmp/interface ethernet1> **send-router-alert**
Router/configure/ip/igmp/interface ethernet1>

applicable models:

All models.

ip igmp startup-query-count

This command configures the number of queries sent out on startup, separated by the Startup Query Interval. (See RFC 236 and RFC 3376.) Use the *no* form of this configuration to return to the default value of Startup Query Count. Note: Specifying a *value* in the *no* form has no effect on the configuration.
When an interface-scoped *startup-query-count* is configured, the configured value is used on the interface. Otherwise, the value used on the interface is the value specified in interface-scoped *robustness*. If an interface-scoped *robustness* is not configured, then the value used on the interface is the value specified in the globally-scoped *startup-query-count*. If globally-scoped *startup-query-count* is not configured, then the value used on the interface is the value of the globally-scoped *robustness*. Finally, if no globally-scoped *robustness* is configured, then the value used on the interface is 2.

syntax:
[no] ip igmp startup-query-count value

parameter	definition
value	An integer between 1 and 10.

example:
The following example configures the globally scoped Startup Query Count to be 10.
Router/configure/ip/igmp/interface ethernet0> **startup-query-count 10**

applicable models:
All models.

ip igmp startup-query-interval

Determines the time between successive General Query messages on startup. (See RFC 3376.) This value is encoded in the Max Resp Code field of Query messages. Use the *no* form of this configuration to return to the default value of Startup Query Interval.

syntax:
[no] ip igmp startup-query-interval time

parameter	definition
time	Number of seconds between 1 and 1024.

example:
The following example configures the default Startup Query Interval to be 1000.
Router/configure/ip/igmp/interface ethernet0> **startup-query-interval 1000**
Router/configure/ip/igmp/interface ethernet0>

applicable models:
All models.

ip igmp version

Specifies the version of IGMP to run. The default version is 3.

syntax:
[no] ip igmp version [1 | 2 | 3]

parameter	definition
1 | 2 | 3	Specifies the version level to run.

example:
The following example configures IGMP version 2 to run on interface ethernet0.
Router/configure/ip/igmp/interface ethernet0> **version 2**
Router/configure/ip/igmp/interface ethernet0>

clear ip igmp groups

Clears all the groups if no optional parameters are specified. With the interface name specified, clears the groups identified from that particular interface. With the group address specified, clears the group regardless of the interface where the information came:

syntax:
[no] clear ip igmp groups [interface <name>] [group-addr <addr>] [source-addr <source-addr>]

parameter	definition
interface name	Specifies the name of the interface, for example ethernet1.
group-addr addr	Specifies the group address.
source-addr addr	Specifies the source address of the group.

example:
The following example clears all groups:
Router> **clear ip igmp groups**

applicable models:
All models.

debug ip igmp all

Enables all levels of diagnostics on IGMP.

syntax:
[no] ip igmp all

example:
The following example enables all debug traces for IGMP:
Router/debug> **ip igmp all**

applicable models:
All models.

debug ip igmp state

Enables all levels of diagnostics on IGMP state-related events.

syntax:
[no] ip igmp state

example:
The following debugs IGMP state events:
Router/debug> **ip igmp state**

applicable models:

All models.

debug ip igmp normal

Enables all levels of diagnostics on IGMP.

syntax:
[no] ip igmp normal

example:
The following debugs normal IGMP traffic:
Router/debug> **ip igmp normal**

applicable models:
All models.

debug ip igmp packet query

Runs diagnostics on IGMP query packets.

syntax:
[no] ip igmp packet query [inbound | outbound]

parameter	definition
inbound	Debugs inbound IGMP traffic.
outbound	Debugs outbound IGMP traffic.

example:
The following example debugs all inbound query packets.
Router/debug> ip igmp packet query inbound

applicable models:
All models.

debug ip igmp packet report

Runs diagnostics on report packets:

syntax:
[no] ip igmp packet report [inbound | outbound]

parameter	definition
inbound	Debugs inbound report packets.
outbound	Debugs outbound report packets.

example:
The following debugs outbound report packets:
Router/debug> **ip igmp packet report outbound**

applicable models:
All models.

debug ip igmp packet leave

Runs diagnostics on all IGMP leave packets.

syntax:
[no] ip igmp packet leave [inbound | outbound]

parameter	definition
inbound	Debugs inbound leave packets.
outbound	Debugs outbound leave packets.

example:
The following example debugs all incoming leave packets:
Router/debug> **ip igmp packet leave inbound**

applicable models:
All models.

show ip igmp groups

Displays IGMP group membership information.

syntax:
show ip igmp groups {all | <interface-name>} [detail]

parameter	definition
all	Displays all IGMP group information
interface name	Displays IGMP group information only for the named interface, for example ethernet1.
detail	Displays in verbose mode.

example:
The following example displays information on all IGMP groups:
Router> **show ip igmp groups all**

applicable models:
All models.

show ip igmp interface all

Displays IGMP interface configuration:

syntax:
show ip igmp interface {all | <interface-name>}

parameter	definition
all	Displays all information for all interfaces.
interface name	Displays configuration information for the named interface, for example ethernet1.

example:
The following example displays all IGMP interface configuration information:

```
Router/configure/ip/igmp>  show ip igmp interface all
IGMP Interface ethernet0 information
interface: ethernet0 10.10.10.1/24, owner: IGMPV3
      Querier: 10.10.10.1 (this system)       Version: 2
      Query Interval: 60 secs
      Query Response Interval: 300 secs
      Last member Query Interval: 30 secs
      Last member Query Count: 7
      Startup Query Interval: 1000 secs
      Startup Query Count: 10
      Send Router Alert: Enabled
      Require Router Alert: Enabled
      Ignore V1 Messages: Disabled
      Ignore V2 Messages: Disabled
      Robustness: 7
      No of Joins on this interface: 0
IGMP Interface ethernet1 information
interface: ethernet1 192.168.123.100/24, owner: PIM-SM
      Querier: 192.168.123.100 (this system)  Version: 3
      Query Interval: 125 secs
      Query Response Interval: 10 secs
      Last member Query Interval: 1 secs
      Last member Query Count: 4
      Startup Query Interval: 31 secs
      Startup Query Count: 2
      Send Router Alert: Enabled
      Require Router Alert: Disabled
      Ignore V1 Messages: Enabled
      Ignore V2 Messages: Enabled
      Robustness: 2
      No of Joins on this interface: 0
Router/configure/ip/igmp>
```

applicable models:

All models.

mtrace

Traces mutlicast routes. Use address 0.0.0.0 for the group address to invoke a weak mtrace. A weak mtrace is one that follows the RPF path to the source, regardless of whether any router along the path has multicast routing table state.

If the first mtrace request fails, three retries are supported with the hop count set with 1, 2 & 3. A '*' character will be displayed in case the mtrace request fails. Use **Ctrl-C** to abort the **mtrace**.

To use this command, IGMP must be enabled in the router and on at least one interface.This command is supported in all modes.

syntax:
mtrace source [destination] [group]

parameter	definition
source	The IP address or host name of a remote system of the multicast-capable source. The default is 127.0.0.1.
group	The IP address or hostname of the group to be traced (default: 224.2.0.1

example:
To trace the route from *192.168.0.0 to 192.168.2.22 via group 225.254.254.254*, enter:
Router> **mtrace 192.168.0.0 192.168.2.22 239.254.254.254**

```
Mtrace from 192.168.2.0 to 192.168.2.22 via group 225.254.254.254
Querying full reverse path...
 1  192.168.2.15 PIM  thresh^ 0  0 ms
 2  192.168.2.7 PIM  thresh^ 0  2 ms
 3  192.168.2.5 PIM  thresh^ 0  674 ms
 4  192.168.2.3 PIM  thresh^ 0  673 ms
 5  192.168.2.2 PIM  thresh^ 0  674 ms
 6  192.168.2.1 PIM  thresh^ 0  673 ms
```

where:
`4 192.168.2.3 – 192.168.2.3` is intermediate router 4 hops away from the destination `PIM thresh^ 0` Multicast Protocol in use on this hop and TTL Threshold.
`674 ms` is the time taken to trace is to be forwarded between hops.
Note—In this release, Maximum Hops and TTL values can not be configured. Maximum Hops is set to 32 and TTL is set to 127 in all mtrace packets as the default.

Multicast Commands

The following section explains the mutlicast mode and interface specific multicast mode commands.

ip multicast

Enters the multicast mode.

syntax:
[no] ip multicast

example:
The following example enables IP multicast mode.
Router/configure/> **ip multicast**
Router/configure/ip/multicast>

applicable models:
All models.

configure ip multicast multipath

Configures the router to do load splitting of multicast traffic over equal cost paths. By default, multipath is enabled with Highest Random Weight (HRW) method. Use the no form of the command to disable multipath. Enabling multipath without specifying the mode, enables multipath with the HRW method. Specifying the mode as `cisco` enables multipath with the Modulo-N Hash method.

syntax:
[no] multipath [mode <cisco> | <hrw>]

parameter	definition
mode cisco	Specifies compatibility with Cisco Systems IOS.
mode hrw	Specifies Highest Random Weight method, the default.

example:
The following command enables compatibility between the Router router and equipment running Cisco IOS.

```
Router/configure/ip/multicast> multipath mode cisco
Router/configure/ip/multicast>
```

The following command enables HRW compatibility.

```
Router/configure/ip/multicast> multipath
Router/configure/ip/multicast>
```

applicable models:

All models.

configure ip multicast static interface <name> group

Configures a static group join on an interface. Specify the optional source address to configure a (S, G) join. Use the no form of this command to delete a previously configured static group join.

syntax:
ip multicast static interface <name> group <group-addr> [<source-addr>]

parameter	definition
interface name	Specifies the name of the interface, for example, ethernet1.
group-addr	Specifies the group address multicast address.
source-addr	Specifies the IP address of the source.

example:
The following example shows how to configure a static group membership.
Router/configure/ip/multicast> **static**
Router/configure/ip/multicast/static>
The following command configures a static group at the multicast address 229.1.1.1 for ethernet0.
Router/configure/ip/multicast/static/interface/ethernet 0> **group 229.1.1.1**
The following command configures a static group at the multicast address 229.1.1.1 for ethernet0 with a source address of 10.1.1.9
Router/configure/ip/multicast/static/interface/ethernet 0> **group 232.0.0.9 10.1.1.9**

applicable models:
All models.

show ip multicast vifs

This command displays all configured virtual interfaces (VIFs) in the system. The VIF index, interface name, IP address and flags for each VIF is displayed. Specifying a VIF name displays only that VIF along with details including traffic statistics.

syntax:
show ip multicast vifs [interface <name>] [pvc <dlci>]

example:
The following example shows information on virtual interfaces.
Router> **show ip multicast vifs**

Total VIFs = 2
register_vif (VIF > 0, 127.0.0.1) Threshold: 1 Flags: REGISTER
ethernet0 (VIF > 1, 192.168.31.193) Threshold: 1 Flags:

applicable models:

All models.

show ip mfc

This command displays the Multicast Forwarding Cache (MFC). Specifying a source and group address displays only the specific entry along with traffic statistics for it.

syntax:
show ip mfc [source <ip-address>] [group <ip-address>]

example:

Router> **show ip mfc**
Router> show ip mfc source 10.1.5.2 group 229.1.1.1

applicable models:

All models.

show ip mroute

This command is used to display the multicast routes in the system. Without any options, it displays both the PIM SM and SSM routing tables. Either one can be displayed by specifying the type to be pim-sm or pim-ssm. Specifying the type as static displays all configured static group joins. The entries can be filtered by specifying group/mask and/or source/mask.

syntax:
show ip mroute [type <pim-sm | pim-ssm | static>] [group <ip-address>] [gmask <ip-mask>] [source <ip-address>] [smask <ip-mask>]

example:
The following example show all the options for this command.

Router> **show ip mroute**
Router> **show ip mroute pim-sm**
Router> **show ip mroute pim-ssm**
Router> **show ip mroute static**
Router> **show ip mroute group 229.0.0.0 gmask 8**
Router> **show ip mroute group 229.0.0.0 gmask 8 source 10.1.0.0 smask 24**

applicable models:

All models.

show ip rpf

This command is used to display RPF (Reverse Path Forwarding) information for a source or RP address. It displays the RPF interface, the unicast route and the associated unicast routing protocol.

syntax:
show ip rpf <ip-address>

example:
The following example shows how to see the reverse path forwarding information for the RP at 201.1.1.99:

Router> **show ip rpf 201.1.1.99**

applicable models:

All models.

clear ip multicast vif-statistics

This command clears traffic statistics for all multicast VIFs in the system. Specifying a VIF name clears statistics only for that VIF.

syntax:
clear ip multicast vif-statistics [interface <name>] [pvc <dlci>]

example:
Router> **clear ip multicast vif-statistics**
Router> **clear ip multicast vif-statistics interface ethernet0**

applicable models:

All models.

clear ip mfc-statistics

Clears traffic statistics associated with an MFC entry.

syntax:
clear ip mfc-statistics [group <ip-address>] [gmask <ip-mask>] [source <ip-address>] [smask <ip-mask>]

example:
Router> **clear ip mfc-statistics group 229.1.1.1 source 10.1.6.9**

applicable models:
All models.

Protocol Independent Multicast

PIM is protocol independent because it depends on existing unicast routes to calculate the reverse path forwarding. PIM has evolved into numerous modes. The following modes are supported in this release:

- Protocol Independent Multicast-Sparse Mode (PIM-SM)
- Protocol Independent Multicast-Source-Specific Multicast (PIM-SSM)

There are two versions of the PIM-SM protocol. PIM-SM version 1 is documented in RFC 2117. PIM-SM version 2 was constructed to address some of the shortcomings of PIM-SM version 1. This release implements only version 2 (RFC 2362).

ip pim

This is a mode command to enter PIM mode. PIM mode configures the PIM router to operate in either sparse mode or dense mode. The default is sparse mode PIM. Use the **no** form of this command to disable PIM globally.

syntax:
[no] pim

example:
To access PIM mode, enter:
Router/configure/ip> **pim**
Router/configure/ip/pim>

applicable models:
All models.

ip pim assert-holdtime

Specifies the number of seconds that Assert state should be maintained in the absence of a refreshing Assert message. When a PIM router receives an Assert message, it modifies the outgoing interface list for a (*,G) or (S, G) entry, as specified by the message. The lifetime of this modification is specified by this command. If another Assert message does not refresh the Assert state before the lifetime expires, then the outgoing interface list reverts to its previous state. Use the *no* form to remove the configured value and return this command to its default value (180 seconds).

syntax:
[no] ip pim assert-holdtime time

parameter	definition
time	The hold time in seconds. Valid Range is: 1—65535 seconds.

example:
The following example configures a global assert-holdtime value of 600.
Router/configure/ip/pim> assert-holdtime 600

applicable models:
All models.

ip pim cbsr

Configures the PIM router as a candidate BSR (CBSR). This is a mode command and will take you to the BSR mode.

syntax:
cbsr

example:
The following example enters the BSR mode.
Router/configure/ip/pim> **cbsr**
Router/configure/ip/pim/cbsr>

applicable models:

All models.

ip pim cbsr interface

Sets the CBSR interface. Use the no form of this command to remove the configured bsr interface.

syntax:
[no] interface name [dlci]

parameter	definition
name	The name of the interface (ethernet0, ethernet1 or bundle name)
dlci	The data link connection identifier of the PV. Valid range is: 16—1022.

example:
The following command sets Ethernet1 as the CBSR interface.
Router/configure/ip/pim/cbsr> interface ethernet1

applicable models:

All models.

ip pim cbsr holdtime

Sets the CBSR holdtime (default: 130). Use the **no** form of this command to set the default value.

syntax:
[no] holdtime time

parameter	definition
time	The BSR hold time. Valid range is: 1—65535.

example:
The following example sets the holdtime to 33 seconds.

```
Router/configure/ip/pim/cbsr> holdtime 33
Router/configure/ip/pim/cbsr>
```

applicable models:

All models.

ip pim cbsr period

set the interval between originating bootstrap messages. (default:60). Use the **no** form of this command to set the default value.

syntax:
[no] period time

parameter	definition
time	The BSR message interval in seconds. Valid range is: 1—65535.

example:
The following example sets the message interval to 1000 seconds:

```
Router/configure/ip/pim/cbsr>  period 1000
Router/configure/ip/pim/cbsr>
```

applicable models:

All models.

ip pim cbsr priority

set the CBSR priority (default: 0). Use the no form of this command to set the default value.

syntax:
[no] priority value

parameter	definition
value	The default CBSR priority. Valid range is 0—255.

example:
The following example sets the priority to 10.
Router/configure/ip/pim/cbsr> **priority 10**
Router/configure/ip/pim/cbsr>

applicable models:
All models.

ip pim crp

A set of routers within a domain are configured as candidate Rendezvous Points (C-RPs). This command configures the PIM router as a candidate RP for the group(s). This is a mode command and enters the CRP mode.

syntax:
crp

example:
To enter the candidate Rendezvous Point mode, enter:
Router/configure/ip/pim> **crp**
Router/configure/ip/pim/crp>

applicable models:

All models.

ip pim crp interface

Set the Candidate RP interface. Use the no form of this command to remove a configured CRP interface.

syntax:
[no] interface name [dlci]

parameter	definition
name	The name (ethernet0, ethernet1 or bundle name) of the interface.
dlci	The data link connection identifier of the PVC. Valid range is 16-1022.

example:
To configure interface ethernet1 as a candidate RP interface, enter:
Router/configure/ip/pim/cbsr> **interface ethernet1**

applicable models:
All models.

ip pim crp group-add

Sets the group to advertise for candidate RP. Use the no form of this command to remove a group for CRP advertisement.

syntax:
[no] group-add address [mask] [priority]

parameter	definition
address	The group IP address used for Candidate RP advertisement.
mask	The subnet mask of the group that is used for the CRP advertisement.
priority	The priority of the group address used for Candidate RP advertisement. The default is the priority of the CRP. Valid Range is: 0—255.

example:
To set the group IP address for CRP advertisements to 224.1.1.0, enter:
Router/configure/ip/pim/crp> **group-add 224.1.1.0**

applicable models:
All models.

ip pim crp holdtime

Set the holdtime advertised in the CRP messages (default: 150 seconds). Use the no form of this command to set the default value.

syntax:
[no] holdtime time

parameter	definition
time	The CRP holdtime in seconds. Valid range is: 1—65535

example:
To set the holdtime to 200 seconds, enter:
Router/configure/ip/pim/crp> **holdtime 200**

applicable models:
All models.

ip pim crp period

Sets the interval at which a CRP will send advertisement messages to the BSR (default: 60 seconds). Use the no form of this command to set the default value.

syntax:
[no] period period

parameter	definition
time	The CRP message interval in seconds. Valid range is: 1—65535

example:
To set the CRP message time interval to 30 seconds, enter:
Router/configure/ip/pim/crp> **period 30**

applicable models:
All models.

ip pim crp priority

set the priority for the CRP (default: 0). Use the no form of this command to set the default value.

syntax:
[no] priority number

parameter	definition
value	Default CRP priority. Valid range is: 0—255.

example:
To set the CRP priority to 45, enter:
Router/configure/ip/pim/crp> **priority 45**

applicable models:
All models.

ip pim hello-holdtime

Sets the period for which the neighbors should wait for Hello messages before expiring the sender's neighbor state (default: 105 seconds). Use the no form of this command to set the default value.

syntax:
[no] hello-holdtime time

parameter	definition
time	The hold time to wait for Hello messages. Valid range is: — 65535

example:
To set the holdtime to 60 seconds, enter:
Router/configure/ip/pim> **hello-holdtime 60**

applicable models:
All models.

ip pim hello-interval

Set the frequency with which the hello messages are to be sent (default: 30 seconds). Use the no form of this command to set the default value.

syntax:
[no] hello-interval time

parameter	definition
time	The length of time to wait before sending out another hello message. Valid range is: 1—65535 seconds.

example:
To set the hello interval time to 145 seconds, enter:
Router/configure/ip/pim> **hello-interval 145**

applicable models:
All models.

ip pim hello-priority

Sets the priority used to determine the Designated Forwarder (default: 1). Use the no form of this command to set the default value.

syntax:
[no] hello-priority priority

parameter	definition
priority	The priority assigned to the Designated Forwarder. Valid range is: 1- 65535.

example:
To set the priority to 15, enter:
Router/configure/ip/pim> **hello-priority 15**

applicable models:
All models.

ip pim interface

Enables PIM interface. This is a mode command and will enter the PIM interface mode. Use the no form of this command to remove PIM configuration from an interface.

syntax:
[no] interface name [dlci <n>]

parameter	definition
name	The name (ethernet0, ethernet1 or bundle name) of the interface.
dlci	The data link connection identifier of the PVC. Valid range is 16-1022.

example:
To configure ethernet1 for PIM, enter:
Router/configure/ip/pim> **interface ethernet1**
Router/configure/ip/pim/interface ethernet1>

applicable models:
All models.

ip pim interface mode

Configures the PIM router for sparse mode for the interface. (Dense mode is not supported in this release).

syntax:
mode [sparse|dense]

example:
To enter PIM SM, enter:
Router/configure/ip/pim/interface ethernet1> **mode sparse**
Router/configure/ip/pim/interface ethernet1>

applicable models:

All models.

ip pim interface hello-holdtime

Sets the period for which the neighbors should wait for Hello messages before expiring the sender's neighbor state (default: 105 seconds). Use the no form of this command to set the default value.

syntax:
[no] hello-holdtime time

parameter	definition
time	The hello hold time in seconds. Valid range is: 0—65535

example:
To configure the hello holdtime to be 35 seconds, enter:
Router/configure/ip/pim/interface ethernet1> **hello-holdtime 35**
Router/configure/ip/pim/interface ethernet1>

applicable models:
All models.

ip pim interface hello-interval

Sets the frequency with which the hello messages are to be sent (default: 30 seconds). Use the no form of this command to set the default value.

syntax:
[no] hello-interval time

parameter	definition
time	The hello interval in seconds. Valid range is: 0— 65535.

example:
To set the holdtime to 90 seconds, enter:
Router/configure/ip/pim/interface ethernet1> **hello-interval 90**
Router/configure/ip/pim/interface ethernet1>

applicable models:
All models.

ip pim interface hello-priority

Sets the priority used to determine the Designated Forwarder (default: 1). Use the no form of this command to set the default value.

syntax:
[no] hello-priority priority

parameter	definition
priority	The hello priority. Valid range is: 1—4294967295

example:
To set the hello priority, enter:
Router/configure/ip/pim/interface ethernet1> **hello-priority 12**
Router/configure/ip/pim/interface ethernet1>

applicable models:
All models.

ip pim interface join-prune-holdtime

Sets the join/prune holdtime advertised in Join/Prune messages (default: 210 seconds). Use the no form of this command to set the default value.

syntax:
[no] join-prune-holdtime time

parameter	definition
time	The join/prune holdtime in seconds. Valid range is: 0—65535.

example:
To set the join-prune hold time to one minute, enter:
Router/configure/ip/pim/interface ethernet1> **join-prune-holdtime 60**
Router/configure/ip/pim/interface ethernet1>

applicable models:
All models.

ip pim interface join-prune-interval

Sets the frequency with which join/prune messages are sent (default: 60 seconds). Use the no form of this command to set the default value.

syntax:
[no] join-prune-interval interval

parameter	definition
interval	The join-prune interval in seconds. Valid range is: 0—65535.

example:
To set the interval to two minutes, enter:
Router/configure/ip/pim/interface ethernet1> join-prune-interval **120**
Router/configure/ip/pim/interface ethernet1>

applicable models:
All models.

ip pim interface boundary

Sets the border of the PIM domain. Use the no form of this command to remove the boundary from an interface.

syntax:
[no] boundary

example:
To set the PIM border for Ethernet 1, enter:
Router/configure/ip/pim/interface ethernet1> **boundary**
Router/configure/ip/pim/interface ethernet1>

applicable models:

All models.

ip pim join-prune-holdtime

Set the join/prune holdtime advertised in Join/Prune messages (default: 210 seconds). Use the no form of this command to set the default value.

syntax:
[no] join-prune-holdtime time

parameter	definition
time	The holdtime (in seconds) that is advertised in join/prune messages. Valid range is: 1— 65535 seconds.

example:
To set the holdtime to 30 seconds, enter:
Router/configure/ip/pim> **join-prune-holdtime 30**

applicable models:
All models.

ip pim join-prune-interval

Sets the frequency with which join/prune messages are sent (default: 60 seconds). Use the no form of this command to set the default value.

syntax:
[no] join-prune-interval interval

parameter	definition
time	The frequency (in seconds) with which join/prune messages are sent. Valid range is: 1— 65535 seconds.

example:
To send messages every five minutes, enter:
Router/configure/ip/pim> join-prune-interval 300

applicable models:
All models.

ip pim mode

Configures the PIM router for sparse mode. Default is sparse-SSM mode.

syntax:
mode [sparse|dense]

parameter	definition
mode	The mode of PIM used on the router. Mode takes two options, sparse to configure sparse mode, and dense which is not supported in this release.

example:
To configure the router for sparse mode PIM, enter:
Router/configure/ip/pim> **mode**

applicable models:
All models.

ip pim mrt-period

Sets the number of seconds to wait between examinations of the MRT (default: 15 seconds). Use the no form of this command to set the default value.

syntax:
[no] mrt-period time

parameter	definition
time	The wait period between checking the MRT. Valid range is: 1—3600.

example:
To check the router table every 15 seconds, enter:
Router/configure/ip/pim> **mrt-period 15**

applicable models:
All models.

ip pim mrt-spt-mult

Used with the *mrt-period* to specify the interval at which the data rate threshold for all S,G entries will be checked for a possible switch to the SP tree. This value, multiplied by the *mrt-period* value, specifies the interval at which the data rate threshold for all (S,G) entries are checked for a possible switch to the shortest path tree. Use the no *ip pim mrt-spt-mult*, to reset the MRT Stale Multiplier to its default (14).

syntax:
[no] ip pim mrt-spt-mult number

parameter	definition
number	an integer between 1 and 100

example:
The following example configures the MRT SPT Mult value to be 25.
Router/configure/ip/pim> mrt-spt-mult 25

applicable models:

All models.

ip pim mrt-stale-mult

Sets the multiple of the mrt-period to time out (S,G) entry (default: 14). Use the no form of this command to set the default value.

syntax:
[no] mrt-stale-mult Number

parameter	definition
number	The mrt-period value multiplier. Valid range is: 1—100.

example:
To set the time out (S, G) entries at 5 times the mrt-period value, enter:
Router/configure/ip/pim> **mrt-stale-mult 5**

applicable models:
All models.

ip pim probe-period

Specifies the number of seconds prior to the RegisterStop timer expiry to send a null Register message to the Rendezvous Point (RP).

syntax:
[no] ip pim probe-interval time

parameter	definition
time	The number of seconds before the RegisterStop timer expires and sends a null Register message to the RP. Valid range is: 1—3600 seconds. The default is 5 seconds.

example:
The following example configures the probe period to 30 seconds.

```
Router/configure/ip/pim> probe-period 30
Router/configure/ip/pim>
```

applicable models:
All models.

ip pim register-suppress-timeout

Specifies the time, in seconds, between receiving a PIM RegisterStop message and allowing Register messages encapsulating multicast data to again be sent. When a router receives a RegisterStop message from a Rendezvous Point (RP) for an (S,G) pair, it must stop sending multicast data encapsulated in Register messages for some period of time. Such a router is said to be "register-suppressed" for the (S,G) pair. This command specifies the number of seconds for which the router remains register-suppressed. A lower value means that the RP receives more frequent bursts of encapsulated multicast data, while a higher value means a longer join latency for new receivers. Use the no form of this command to return this to its default value of 60 seconds.

syntax:
[no] register-suppress-timeout time

parameter	definition
Time	The register suppress timeout in seconds. Valid range is: 1—3600.

example:
The following example configures the Register Suppression Timeout to be 70 seconds.
Router/configure/ip/pim> register-suppress-timeout 70

applicable models:

All models.

ip pim rp

Sets the static Rendezvous Point router address. Use the no form of this command to remove the configured static RP address.

syntax:
[no] rp address [group-address] [group-mask]

parameter	definition
address	The static RP IP address.
group-address	The multicast group IP address.
group-mask	The multicast group subnet mask.

example:
To set the RP static IP address to 10.10.1.1, enter:
Router/configure/ip/pim> **rp 10.10.1.1**

applicable models:
All models.

ip pim rp-switch-immediate

Causes a Rendezvous Point (RP) to initiate a switch to the Shortest Path (SP) tree for (S,G) upon receipt of the first Register message encapsulating data from source S. The PIM-SM protocol allows an RP or a Designated Router (DR) to switch from receiving data from a source S sent to a group G via the RP tree, to receiving data from the SP tree. If this command is not configured, then an active RP will initiate a switch to the SP tree when the traffic rate exceeds a threshold. If this option is set, then any other options related to the default method of switching will have no effect. Once this option is configured, the only way to return to the default mode is to deconfigure it using the *no* form. This parameter is on by default.

syntax:
[no] rp-switch-immediate

example:
To turn on this feature, enter:
Router/configure/ip/pim> **rp-switch-immediate**

applicable models:

All models.

ip pim ssm-range

Configures a range of multicast addresses to be treated as SSM group addresses. (Default is 232/8).

syntax:
ssm-range group-address group-mask

parameter	definition
group-address	The multicast group IP address.
group-mask	The multicast group subnet mask.

example:
To configure the range of multicast address starting at 224.20.12.1/24 to be treated as SSM group addresses, enter:
Router/configure/ip/pim> **ssm-range 224.20.12.1 24**

applicable models:
All models.

ip pim threshold-dr

Specifies the threshold, in kilobytes per second, for a Designated Router (DR), which, when exceeded for an (S,G) pair, triggers a switch to the shortest path tree. If the *dr-switch-immediate* option is configured, configuring this option has no effect. Use the no form of this command to return the value to the default of 1 KBps.

syntax:
[no] ip pim threshold-dr KBps

parameter	definition
KBps	The threshold in kilobytes per second. Valid range is: 1—4294967.

example:
To configure this router such that the data from S addressed to G must exceed an average of 1024 KBytes per second before an SPT switch is initiated, enter:
Router/configure/ip/pim> threshold-dr 1024

applicable models:
All models.

ip pim threshold-rp

Specifies the threshold, in kilobytes per second, for a Rendezvous Point (RP), which, when exceeded for an (S,G) pair, initiates a switch to the Shortest Path (SP) tree. If the *dr-switch-immediate* option is configured, then configuring this option has no effect. Use the no form of this command to return the value to the default of 1 KBps.

syntax:
[no] ip pim threshold-rp KBps

parameter	definition
KBps	The threshold in bps. Valid Range is: 1—4294967.

example:
To configure this router such that the data from S addressed to G must exceed an average of 1500 KBytes per second before an SPT switch is initiated, enter:
Router/configure/ip/pim> **threshold-rp 1500**

applicable models:
All models.

ip pim whole-packet-checksum

Specifies that checksums in Register messages should be calculated over the entire encapsulated data packet, rather than just over the Register message header.
If *ip pim whole-pkt-checksum* is not specified, it is the same as specifying no ip pim whole-pkt-checksum.

syntax:
[no] ip pim whole-pkt-checksum

example:
To specify that the message checksum will be calculated over the entire encapsulated packet, rather than just over the Register message header, enter:
Router/configure/ip/pim> **whole-packet-checksum**

applicable models:

All models.

show ip pim global

Displays PIM-SM configuration information.

syntax:
global

example:
To display PIM global configuration settings, enter:

```
Router/configure>  show ip pim global
PIM: Enabled
      Mode: Sparse
      Timers:
            Hello Interval: 145
            Hello Hold Time: 60
            Hello Priority: 15

            Join/Prune Interval: 300
            Join/Prune Hold Time: 30
            Assert Hold Time: 200
            Probe Period: 15
            Register Suppress Timeout: 90
            MRT Interval: 15
            MRT SPT Multiplier : 10
            MRT Stale Multiplier: 5
      Thresholds:
            Threshold DR: 2400
            Threshold RP: 1500
      RP Switch Immediate: enabled
      DR Switch Immediate: enabled
      Whole packet checksum: enabled
      SSM Range: 224.20.12.1 24
Router/configure>
```

applicable models:
All models.

show ip pim interface

Displays PIM interface information.

syntax:
interface [detail]

parameter	definition
interface	Specifies the interface. Use the keyword all to display information about all the PIM interfaces.
detail	The detail mode of display.

example:
To display information for all interfaces, enter:

```
Router/configure> show ip pim interface all
Address         Interface     Nbr   Hello Hello   Hello JP    DR
                              Count Intvl HldTime Pri   Intvl
--------------- ------------- ----- ----- ------- ----- ----- -----------
Router/configure>
```

applicable models:
All models.

show ip pim neighbors

Displays information on PIM neighbors.

syntax:
neighbors

example:
To display information on PIM neighbors, enter:

```
Router/configure> show ip pim neighbors
Neighbor       Interface    Uptime   Expires         Hello Priority
-------------- ------------ -------- --------------- --------------
Router/configure>
```

applicable models:
All models.

show ip pim rp

Displays PIM Rendezvous Point information.

syntax:
rp

example:
To display RP information, enter:

```
Router/configure>  show ip pim rp
      Group/Mask         RP
      ------------------ ---------------
      224.0.0.0/4        10.10.1.1
Router/configure>
```

applicable models:

All models.

show ip pim rp-set

Displays PIM RP-set information.

syntax:
rp-set

example:
To view RP-set information, enter:

```
Router/configure> show ip pim rp-set
Group/mask       Src/RP         Pri Uptime    Expires
----------       ------         --- ------    -------
224/4            10.10.1.1        1 Static RP
  Dependencies:  None

Router/configure>
```

applicable models:
All models.

show ip pim statistics

Displays PIM statistics.

syntax:
statistics

example:
To view PIM counters, enter:

```
Router/configure>  show ip pim statistics
PIM Statistics:
                Total PIM msgs recvd  0 (0 bytes)
                Recvd msgs too short  0
              Recvd msgs bad checksum  0
              Recvd msgsg bad version  0
                 Recvd register msgs  0 (0 bytes)
            Recvd registers wrong iif  0
                Recvd bad registers  0
                 Sent register msgs  0 (0 bytes)
Router/configure>
```

applicable models:
All models.

show ip pim timers

Displays PIM timer information.

syntax:
timers

example:
To display PIM timer information, enter:

```
Router/configure> show ip pim timers
PIM Timers:
      Hello Interval: 145
      Hello Hold Time: 60
      Hello Priority: 15

      Join/Prune Interval: 300
      Join/Prune Hold Time: 30
      Assert Hold Time: 200
      Probe Period: 15
      Register Suppress Timeout: 90
      MRT Interval: 15
      MRT SPT Multiplier : 10
      MRT Stale Multiplier: 5
Router/configure>
```

applicable models:
All models.

show ip pim bsr-info

Displays PIM BSR information.

syntax:
bsr-info

example:
To examine PIM BSR statistics, enter:

```
Router/configure/ip/pim>  show ip pim bsr-info
Candidate BSR Information
-------------------------
Candidate BSR Status: Disabled
Candidate BSR Interface: NOT CONFIGURED
      Candidate BSR Priority: 45
      Candidate BSR Period: 30

      Candidate BSR Hold Time: 2048
      Candidate BSR Admin Scope: Disabled

No BSR's
Router/configure/ip/pim>
```

applicable models:
All models.

clear ip pim statistics

Resets all PIM statistical information.

syntax:
statistics

example:
To reset PIM counters, enter:
Router> **clear ip pim statistics**

applicable models:
All models.

12
IPSEC COMMANDS

Use the **IPSec** commands to configure a Router system with IP security protocol encryption. This is the recommended protection for networking communications.

crypto

Enters crypto mode and provides access to commands used in Router's IPSec VPN feature.

syntax:

configure crypto

example:

This example show how to enter crypto mode:

Router> **configure**
Router/configure> **crypto**

applicable models:

All models.

crypto ike policy

Defines an Internet Key Exchange (IKE) policy for a dynamic ISAKMP SA.

syntax:

[no] configure crypto ike policy <policy-name><peer-address>

parameter	definition
<policy-name>	Specifies an IKE policy name. 1-8 characters.
<peer-address>	:Peer IP address.

example:

Router> **configure term**
Router/configure> **crypto**
Router/configure/crypto> **ike policy ToNetSc1 100.1.1.2**

applicable models:

All models.

crypto ike policy <policy-name> <peer-address> local-address <local-ip-address>

Configures a local address for an IKE policy. When executed, this command creates a default IKE policy proposal is created with Preshared Key, 3DES, SHA1, and DH-group1. The **no** form of command does not exist.

syntax:

configure crypto ike policy <policy-name><peer-address> local-address <local-ip-address>

parameter	definition
<local-ip-address>	Any valid local IP address. If no local interface is configured with this address, command execution will be failed. Default: None.

example:

Router> configure term
Router/configure> **crypto**
Router/crypto/ike> **ike policy ToNetSc1 100.1.1.2**
Router/crypto/ike/policy ToNetSc1> **local-address 100.1.1.1**

applicable models:

All models.

crypto ike policy <policy-name> <peer-address> local-id <id-type> <data>

Configures IPSec identifiers for the host that will be used in the identification payload during IKE negotiation. Use the **no** form to configure the local-id to the default values (id-type as the IP address and data as the local address).

syntax:

[no] configure crypto ike policy <policy-name> <peer-address> local-id | domain-name | email-id] <data>

parameter	definition
[domain-name]	Specifies a fully-qualified domain name string like `Router.com`.
[email-id]	Specifies a fully-qualified user name string, as per RFC 822, like `name@Router.com`.
<data>	The identifier according to the type specified. Default: Type is address and id is the local interface IP address specified.

example:

Router> **configure term**
Router/configure> **crypto**
Router/crypto/ike> **ike policy ToNetSc1 100.1.1.2**
Router/crypto/ike/policy ToNetSc1100.1.1.2> **local-id email-id user@alcatel.com**

applicable models:

All models.

crypto ike policy <policy-name> <peer-address> remote-id <id-type> <data>

Configures the IPSec peer identifier that will participate in the IKE negotiation.

Use the **no** form to configure the local-id to the default values (id-type as the IP address and data as the local address).

syntax:

[no] configure crypto ike policy <policy-name> <peer-address> remote-id remote-id-type data

parameter	definition
[domain-name]	Specifies a fully-qualified domain name string like [Router.com.]
[emailid]	Specifies a fully-qualified user name string, as per RFC 822, like [name@Router.com.]
<data>	The identifier according to the type specified.
Default	Type is address and id is the peer address specified with [peer] command.

example:
Router> **configure term**
Router/configure> **crypto**
Router/crypto/ike> **ike policy ToNetSc1 100.1.1.2**
Router/crypto/ike/policy ToNetSc1100.1.1.2> **remail-id email-id user@alcatel.com**

applicable models:

crypto ike policy <policy-name> <peer-address> mode <ike-mode>

Configures the IKE policy mode. To reset the mode to the default value, use the **no** form.

syntax:

[no] configure crypto ike policy <policy-name><peer-address> mode [**aggressive | main**]

parameter	definition
aggressive	It takes 3 messages to establish a security association, has less negotiation power, and does not provide identity protection.
main	It takes 6 messages, in three peer-to-peer exchanges to establish a security association. These three steps include IKE SA negotiation, Diffie-Hellman key exchange, and peer authentication. It provides identity protection. Default: main mode.

example:

Router> **configure crypto**
Router/crypto/ike> **ike policy ToNetSc1 100.1.1.2**
Router/crypto/ike/policy ToNetSc1> **mode aggressive**

applicable models:

All models.

crypto ike policy <policy-name> <peer-address> pfs

Configures Perfect Forward Secrecy feature for an IKE policy. To reset the mode to the default value, use the **no** form.

syntax:

[no] configure crypto ike policy <policy-name><peer-address> pfs

parameter	definition
pfs	Enable/disable PFS feature. Default: disabled.

example:

Router> **configure crypto**
Router/crypto/ike> **ike policy ToNetSc1 100.1.1.2**
Router/crypto/ike/policy ToNetSc1> **pfs**

applicable models:

All models.

crypto ike policy <policy-name> <peer-address> exchange-type <exchange-type>

Allows an user to configure IKE exchange type. Types could be Initiator only, Responder only, Both.

syntax:

[no] configure crypto ike policy <policy-name><peer-address> exchange-type <**initiator-only | responder-only | both**>

parameter	definition
initiator-only	This policy does not respond to IKE negotiation initiated by another party. This type is not supported if the IKE policy is configured for main mode and Preshared Key is configured as the authentication mode.
responder-only	This policy does not initiate IKE to any other party, but will respond to the ike initiated by others.
both	This policy is capable of responding to IKE negotiation, also it can initiate ike negotiation process. Default: both

example:
Router> c**onfigure crypto**
Router/crypto/ike> **policy ToNetSc1 100.1.1.2**
Router/crypto/ike/policy ToNetSc1> **exchange-type initiator-only**

applicable models:

All models.

crypto ike policy <policy-name> <peer-address> key <key-string>

Define a preshared key for the IKE policy. It is valid only when the proposal configured has the authentication method as pre-shared-key.

syntax:

[no] configure crypto ike policy <policy-name><peer-address> key <key-string>

parameter	definition
<key-string>	The ASCII key value for the preshared secret.

example:

Router> **configure crypto**
Router/crypto/ike> **policy ToNetSc1 100.1.1.2**
Router/crypto/ike/policy ToNetSc1> **key hello**

applicable models:

All models.

crypto ike policy <policy-name> <peer-address> proposal < proposal-priority >

Define an IKE proposal for a dynamic ISAKMP SA.

syntax:

[no] configure crypto ike policy <policy-name><peer-address> proposal <proposal-priority>

parameter	definition
<proposal-priority>	Valid values are: 1-5.

example:

Router> **configure crypto**
Router/configure/crypto> **ike policy toOpal 100.1.1.2**
Router/crypto/ike/policy> **proposal 1**

applicable models:

All models.

crypto ike policy <policy-name> <peer-address> proposal < proposal-priority > hash-algorithm <algorithm>

Configures the IKE authentication algorithm for a given proposal. To reset the authentication algorithm to the default value, use the **no** form.

syntax:

[no] configure crypto ike policy <policy-name> <peer-address> proposal <proposal priority> hash-algorithm [md5|sha1]

parameter	definition
md5	.Produces a 128-bit message digest. RFC 1321.
sha1	Produces a 160-bit message digest. Default: sha1

example:

Router> **configure crypto**
Router/crypto> **ike policy ToNetSc1 100.1.1.2**
Router/crypto/ike/policy> **proposal 1**
Router/crypto/ike/policy/proposal ToNetSc1> **hash-algorithm md5**

applicable models:

All models.

crypto ike policy <policy-name> <peer-address> proposal < proposal-priority > authentication-method <authentication-method>

Configures the IKE authentication method to authenticate the peers.

syntax:

[no] configure crypto ike policy <policy-name> proposal <proposal-priority> authentication-method [**pre-shared-key | dss-signature**]

parameter	definition
[preshared| dss-signature]:	Methods that authenticate the peers. Default: pre-shared-key RSA and DSS are not supported in this release.

example:

Router> **configure crypto**
Router/crypto> i**ke policy toOpal 100.1.1.1**
Router/crypto/ike/policy toOpal 100.1.1.1> **proposal 1**
Router/crypto/ike/policy toOpal 100.1.1.1> **authentication-method dss-signature**

applicable models:

All models.

crypto ike policy <policy-name> <peer-address> proposal < proposal-priority > dh-group <group>

Configures the IKE Diffie-Hellman group for key exchange between peers

syntax:

[no] configure crypto ike policy <policy-name> <peer-address> proposal <proposal priority> dh-group [**group1 | group2 | group5**]

parameter	definition
group1	768-bit. RFC 2409. Type of Diffie-Hellman prime modulus group when performing key exchange. Default is group1.
group2	1024-bit. RFC 2409 Type of Diffie-Hellman prime modulus group when performing key exchange.
group5	1536-bit. Type of Diffie-Hellman prime modulus group that IKE will use for the PFS key exchange. This is the highest level of security and requires more process time than group 1 and group 2.

example:

Router> **configure crypto**
Router/crypto> ike policy toOpal 100.1.1.1
Router/crypto/ike/policy toOpal 100.1.1.1> **proposal 1**
Router/crypto/ike/policy toOpal 100.1.1.1/proposal> **dh-group group1**

applicable models:

All models.

crypto ike policy <policy-name> <peer-address> proposal < proposal-priority > encryption-algorithm <algorithm>

Configures the IKE encryption algorithm for a proposal.

syntax:

[no] configure crypto ike policy <policy-name> <peer-address> proposal <proposal-priority> encryption-algorithm **[des-cbc | 3des-cbc | aes128-cbc | aes192-cbc | aes256-cbc]**

parameter	definition
des-cbc	.Encryption algorithm. 8-byte (64-bit) block size. 56-bit (8-byte (64-bit) input with parity bits removed) key size. Default.
3des-cbc	Encryption algorithm. 8-byte (64-bit) block size. 168-bit (24-bytes (192-bit) input with parity bits removed) key size.
aes-cbc	Encryption algorithm. 8-byte (64-bit) block size. 56-bit (64-bit input with parity bits removed) key size.

example:

Router> **configure crypto**
Router/crypto> **ike policy toOpal 100.1.1.1**
Router/crypto/ike/policy toOpal 100.1.1.1l> **proposal 1**
Router/crypto/ike/policy toOpal 100.1.1.1/proposal> **encryption-algorithm 3des-cbc**

applicable models:

All models.

crypto ike policy <policy-name> <peer-address> proposal < proposal-priority > hash-algorithm <algorithm>

Configures the IKE hash algorithm for a proposal.

syntax:

[no] configure crypto ike policy <policy-name> <peer-address> proposal <proposal-priority> hash-algorithm **[md5 | sha1]**

parameter	definition
md5	Produces a 128-bit message digest per RFC 1321.
sha1	Produces a 160-bit message digest. This is the default.

example:

Router> **configure crypto**
Router/crypto> **ike policy toOpal 100.1.1.1**
Router/crypto/ike/policy toOpal 100.1.1.1l> **proposal 1**
Router/crypto/ike/policy toOpal 100.1.1.1/proposal> **hash-algorithm md5**

applicable models:

All models.

crypto ike policy <policy-name> <peer-address> proposal < proposal-priority > lifetime [seconds <secs> | kilobytes <kb>]

Configures lifetime of the IKE SA. When the SA expires, it is replaced by a new negotiated SA or terminated. To reset the lifetime to the default value, use the **no** form.

syntax:

[no] configure crypto ike policy <policy-name> <peer-address> proposal <proposal priority> lifetime [seconds <secs> | kilobytes <kb>]

parameter	definition
seconds <secs>	Specifies the number of seconds the IKE SA will live before expiring. The default is 24 hrs.
kilobytes <kb>	Specifies the volume of traffic (in kilobytes) that can pass between IPSec peers using the given IKE SA before that SA expires. The default is unlimited bytes.

example:

Router> **configure crypto**
Router/crypto > **ike toOpal 100.1.1.1**
Router/crypto/ike/policy toOpal 100.1.1.1> **proposal 1**
Router/crypto/ike/policy toOpal 100.1.1.1/proposal> **lifetime seconds 60**

applicable models:

All models.

crypto ipsec

Configures IPSec. The statements are explained separately.

syntax:

[no] configure crypto ipsec

example:

Router> **configure crypto**
Router/crypto> **ipsec**

applicable models:

All models.

crypto ipsec policy

Define an IPSec policy for a dynamic IPSec SA.

syntax:

[no] configure crypto ipsec policy <policy-name>

parameter	definition
<policy-name>	Specifies an IPSec policy name. 1-8 characters.

example:

Router> **configure crypto**
Router/crypto/ipsec> **policy ToNetSc1 100.1.1.2**

applicable models:

All models.

crypto ipsec policy <peer-address> match address

Configures the IP stream to be applied to IPSec. Delete the policy to remove the addresses. The 'no' form of this command is not supported.

syntax:

configure crypto ipsec policy <policy-name><peer-address> match address

parameter	definition
match address	Configures this command to assign addresses to match with the policy. This specifies the traffic that will be protected by the IPSec policy. Matches are: <source-start-ip> <source-netmask> - the source starting IP address and subnet mask <dest-start-ip> <dest-netmask> - the destination starting IP address and subnet mask [source-end-ip <ipaddress>] - the source ending IP address [dest-end-ip <ipaddress>] - the destination ending IP address [protocol <protocol>] - the protocol [sport <port-val>] - the source port number [dport <port-val>] - the destination port number

example:

Router> **configure crypto**
Router/crypto> i**psec policy ToNetSc1 100.1.1.1**
Router/crypto/ipsec/policy ToNetSc1> **match address 10.1.1.0 24 20.1.1.0 24**

applicable models:

All models.

crypto ipsec policy <peer-address> enable

Enable/disable ipsec policy. Use 'no' form of command to disable ipsec policy.

syntax:

[no] configure crypto ipsec policy <policy-name><peer-address> enable

parameter	definition
enable	Enables a disabled ipsec policy. Default: all ipsec policies are enabled by default.

example:

Router> **configure crypto**
Router/crypto> **ipsec policy ToNetSc1 100.1.1.1**
Router/crypto/ipsec/policy ToNetSc1> **no enable**

applicable models:

All models.

crypto ipsec policy <peer-address> pfs-group <group>

Specify that IPSec should use perfect forward secrecy during establishment of new security associations. To disable use of PFS (the default), use the **no** form.

syntax:

[no] configure crypto ipsec policy <policy-name><peer-address> pfs-group [**group1 | group2| group5**]

parameter	definition
group1	768-bit. Type of Diffie-Hellman prime modulus group that IKE will use for the PFS key exchange.
group2	1024-bit. Type of Diffie-Hellman prime modulus group that IKE will use for the PFS key exchange.
group5	1536-bit. Type of Diffie-Hellman prime modulus group that IKE will use for the PFS key exchange. This is the highest level of security and requires more process time than group 1 and group 2

example:
Router> **configure crypto**
Router/crypto> **ipsec policy ToNetSc1 100.1.1.1**
Router/crypto/ipsec/policy ToNetSc1> **pfs-group group1**

applicable models:

All models.

crypto <security-group-name> ipsec policy <name> <peer-address> proposal <proposal-priority> [protocol]

Defines an IPSec proposal for a dynamic SA. In case multiple proposals are configured, all of them will be sent in the SA payload in a logical OR manner, and in the order they are specified by the proposal priority. The protocol value defaults to ESP if it is not explicitly specified.

Whenever a new proposal is created, the proposal parameters are set to the following defaults:

- For ESP—3DES, SHA1, and tunnel mode
- For AH—SHA1 and tunnel mode

syntax:

[no] configure crypto ipsec policy <name> <peer-address> proposal <proposal priority>

[ah | esp]

parameter	definition
<proposal priority>	The range is 1-5.

example:

Router> **configure crypto**
Router/crypto> **policy ToNetSc1 100.1.1.1**
Router/crypto/ipsec/policy ToNetSc1> **proposal 1**

applicable models:

All models.

crypto ipsec policy <name> <peer-address> proposal < proposal-priority > [protocol] hash-algorithm <algorithm>

Configures the IPSec authentication algorithm for a given proposal. To reset the authentication algorithm to the default value, use the **no** form.

syntax:

[no] configure crypto ipsec policy <name> <peer-address> proposal <priority> [protocol] hash-algorithm [**md5-hmac | sha1-hmac**]

parameter	definition
md5-hmac	.Produces a 128-bit message digest. RFC 1321 + RFC 2085.
sha1-hmac	Produces a 160-bit message digest. Default: sha1-hmac.

example:

Router> **configure crypto**
Router/crypto > **ipsec policy toOpal 100.1.1.1**
Router/crypto/ike/policy toOpal 100.1.1.1> **proposal 1**
Router/crypto/ike/policy toOpal 100.1.1.1> **hash-algorithm md5-hmac**

applicable models:

All models.

crypto ipsec policy <name> <peer-address> proposal < proposal-priority > [protocol] encryption-algorithm <algorithm>

Configures the IPSec encryption algorithm for a proposal.

syntax:

[no] configure crypto ipsec policy <name><peer-address> proposal <proposal-priority> [protocol] encryption - algorithm [**des-cbc** | **3des-cbc** | **aes128-cbc| aes192-cbc| aes256-cbc**]

parameter	definition
des-cbc	8-byte (64-bit) block size. 56-bit (8-byte (64-bit) input with parity bits removed) key size.
3des-cbc	8-byte (64-bit) block size. 168-bit (24-bytes (192-bit) input with parity bits removed) key size. The default.
aes-128cbc	128-bit Advanced Encryption Standard.
aes-192cbc	192-bit Advanced Encryption Standard.
aes-256cbc	256-bit Advanced Encryption Standard.

example:

Router> **configure crypto**
Router/crypto> **ipsec policy toOpal 100.1.1.1**
Router/crypto/ipsec> **proposal 1**
Router/crypto/ipsec/proposal toOpal> **encryption-algorithm 3des-cbc**

applicable models:

All models.

crypto ipsec policy <name> <peer-address> proposal < proposal-priority > [protocol] lifetime [seconds <secs> | kilobytes <kb>]

Configures lifetime of the IPSec SA. When the SA expires, it is replaced by a new negotiated SA or terminated. To reset the lifetime to the default value, use the **no** form.

syntax:

[no] configure crypto ipsec policy <name> proposal < proposal-priority > [protocol] lifetime [seconds <secs> | kilobytes <kb>]

parameter	definition
seconds <secs>	.Specifies the number of seconds the IPSec SA will live before expiring. The default is 1 hr.
kilobytes <kb>	Specifies the volume of traffic (in kilobytes) that can pass between IPSec peers using the given IPSec SA before that SA expires. The default is unlimited bytes

example:

Router> **configure crypto**
Router/crypto> **ipsec policy toOpal 100.1.1.1**
Router/crypto/ipsec policy toOpal> **proposal 1**
Router/crypto/ipsec policy toOpal > **lifetime seconds 60**

applicable models:

All models.

crypto ipsec policy <name> <peer-address> proposal < proposal-priority > [protocol] mode <ipsec-mode>

Configures ipsec mode. To un-configure mode to default mode, use the **no** form.

syntax:

[no] configure crypto ipsec policy <name> <peer-address> proposal < proposal-priority > [protocol] mode [**tunnel** | **transport**]

parameter	definition
tunnel:	Tunnel mode configuration. In tunnel mode the ip header of the packet will be encapsulated into a new ip header with routable destination IP address. The protection is offered for the complete packet. The default.
transport	Transport mode configuration. In transport mode, the old IP address will be retained and the hash (incase of AH) will be generated over the payload and will be delivered to the peer. The protection is offered only for the pay load.

example:

Router> **configure crypto**
Router/crypto> **ipsec toOpal 100.1.1.1**
Router/crypto/ipsec> **proposal 1**
Router/crypto/ipsec/proposal 1> **mode transport**

applicable models:

All models.

show crypto ike policy [<policy-name>] [<proposal priority>] [detail]

View the configured IKE proposal(s).

syntax:

show crypto ike policy [<policy-name>] [<proposal priority>] [detail]

parameter	definition
[all | <proposal priority(s)>]	Use all the view all the IKE proposals configured. Use a specific proposal priority for viewing that proposal configuration. [detail] option specifies that user is requested for detail information of the policy.

example:
Router> **show crypto ike policy toOpal 1**

applicable models:

All models.

show crypto ike sa [<policy-name>] [detail]

View the configured IKE sa(s).

syntax:

show crypto ike sa [all | <policy-name>] [detail]

parameter	definition
[all | <proposal priority(s)>]	Use all the view all the IKE policies configured. Use a specific policy name for viewing that policy configuration.

example:

Router> **show crypto ike sa all**

applicable models:

All models.

show crypto ipsec policy [<policy-name> [<proposal-priority>] [detail]

View the configured IPSec proposal(s).

syntax:

show crypto ipsec policy [<policy-name>] [<priority>] [detail]

parameter	definition
[all | <proposal priority(s)>]	Use all the view all the IPSec proposals configured. Use a specific proposal priority for viewing that proposal configuration.

example:
Router> **show crypto ipsec policy toOpal 1**

applicable models:
All models.

show crypto ipsec sa [all | <policy-name>]

View the configured/established IPSec SA(s).

syntax:

show crypto ipsec sa [all | <policy-name>]

parameter	definition
[all | <policy-name(s)>]	Use all the view all the IPSec SAs configured/established. Use a specific sa name for viewing that SA configuration. Specifying a destination address will show all the SAs that are established with that peer.

example:

Router> **show crypto ipsec sa all**

applicable models:

All models.

show crypto dynamic ike policy [policy-name] [proposal-priority] [detail]

View the configured remote access IKE proposal(s).

syntax:

show crypto dynamic ike policy [policy-name] [proposal-priority] [detail]

parameter	definition
[all | <proposal priority(s)>]	Use all the view all the IKE proposals configured. Use a specific proposal priority for viewing that proposal configuration. [detail] option specifies that user is requested for detail information of the policy.

example:
Router> **show crypto dynamic ike policy toOpal 1**

applicable models:

All models.

show crypto dynamic ike sa [<policy-name>] [detail]

View the configured/established remote access IKE sa(s).

syntax:

show crypto dynamic ike sa [all | <policy-name>] [detail]

parameter	definition
[all | <proposal priority(s)>]	Use all the view all the IKE policies configured. Use a specific policy name for viewing that policy configuration.

example:

Router> **show crypto dynamic ike sa all**

applicable models:

All models.

show crypto ipsec template <modecfg-group | user-group > <group-name> [<policy-name>][<proposal-priority>] [detail]

View the configured IPSec modeconfig or user group templates(s).

syntax:

show crypto ipsec policy [<policy-name>] [<priority>] [detail]

parameter	definition
<modecfg-group| user-group-name>	This indicates which ipsec template is requested by the user.
[all | <proposal priority(s)>	Use all the view all the IPSec proposals configured. Use a specific proposal priority for viewing that proposal configuration.

example:

Router> **show crypto dynamic ipsec template user-group accounts toOpal 1**

applicable models:

All models.

show crypto dynamic ipsec sa <user-group | modecfg-group> <group-name> [all | <policy-name>]

View the configured/established remote access IPSec SA(s).

syntax:

show crypto dynamic ipsec sa <user-group | modecfg-group> <group-name> [all | <policy-name>]

parameter	definition
<modecfg-group| user-group-name>	This indicates which ipsec template is requested by the user.
[all | <policy-name(s)>]	Use all the view all the IPSec SAs configured/established. Use a specific sa name for viewing that SA configuration. Specifying a destination address will show all the SAs that are established with that peer.

example:

Router> **show crypto dynamic ipsec sa user-group accounts all**

applicable models:

All models.

show crypto dynamic modecfg-group

View the configured modeconfig group(s).

syntax:

show crypto dynamic modecfg-group [group-name] [detail]

parameter	definition
< [all | <group-name(s)>]	Use all the view all the configured modeconfig groups for a given security group. Use a specific group name for viewing that group configuration.

example:
Router> **show crypto dynamic modecfg-group Router-mdcfg detail**

applicable models:

All models.

show user-group [user-group-name] [detail]

View the user group configuration(s).

syntax:

show user-group [user-group-name] [detail]

parameter	definition
[all | <user-group-name>]	All specifies to display all user information for a given security group. The detail option specifies that user is requested for detail information of user group.

example:

Router> **show user-group accounts-users detail**

applicable models:

All models.

clear crypto ike sa [<policy-name>]

Clear established IKE SAs.

syntax:

clear crypto ike sa [<policy-name>]

example:

Router> **clear crypto ike sa all**

applicable models:

All models.

clear crypto ipsec sa [<policy-name>]

Clear established IPSec SAs.

syntax:

clear crypto ipsec sa [<policy-name>]

example:

Router> **clear crypto ipsec sa all**

applicable models:

All models.

debug crypto ike

Use this command to display IKE events and debug information. Use the **no** form to disable the debugging output.

syntax:

[no] debug crypto ike

example:
Router> **debug crypto ike**

applicable models:

All models.

debug crypto ipsec <module>

Use this command to display IPSec events and debug information. Use the **no** form to disable the debugging output.

syntax:

[no] debug crypto ipsec

example:

Router> debug crypto ipsec all

applicable models:

All models.

13
PASSWORD

Use the **password** command change passwords on a Router system. The new password is effective at the next login session.

The **password** command is as follows:

Password Command

password

password

This command changes the password on a Router system.

The new password is effective at the next login session. Passwords are limited to 10 characters. If the new password exceeds 10 characters, the password will be ignored and not take effect. An error message is not displayed when the character limit is exceeded.

parameter	definition
old password	Your current password
	Use 3 - 10 characters.
new password	Your new password
	Use 3 - 10 characters.
	You must enter the new password twice to confirm the change.

syntax:

password < old password > < new password > < new password >

example:
Router> **password**
name: **Router**
old password: **john**
new password: **jsmith**
re-enter password: **jsmith**
password has been changed

In this example, the password changes from **john** to **jsmith.** After you enter the **password** command, the system prompts you for the current and new passwords, one at a time**.** Type the new password twice, as prompted, to confirm the change.

applicable models:

All models.

14 PING

Use the **ping** command to verify connectivity between a Router system and other network hosts.

The **ping** command is as follows:

Ping Command

ping

NOTE: You can stop pinging using the key combination, **Ctrl+C**.

ping

This command verifies connectivity between a Router system and other network hosts.

parameter	definition
dipaddress	Destination IP address
sipaddress	Source IP address
pingcnt	Number of ping packets to send
	The range is 1 - 65535; the default is 5.
pktsize	Packet size
	The range is 36 - 10000; the default is 64.
timeout	Timeout in seconds
	The range is 1 - 3600; the default is 5.
fillpattern	ICMP data fill pattern
	The range is 0 - 255; the default is 255.
sweepmin	Sweep minimum packet size
	The range is 36 - 10000; the default is 16.
sweepmax	Sweep maximum packet size
	The range is 36 - 10000; the default is 1016.
tos	Type of service
	The range is 0 - 255; the default is 0.
dfbit	Turn dfbit on in IP header.
off	Default value, turns dfbit off.
on	Turn dfbit on.
options	Ping options
none	None
	The default is none.
verbose	Verbose mode
validate	Validate reply data

syntax:

ping dipaddress <*ip address* > [*sipaddress* < *ipaddress* >] [pingcnt < n >] [pktsize < n >] [timeout < n >] [fillpattern < n >] [sweepmin < n >] [sweepmax < n >] [tos < n >] [dfbit < on | off >] [options < none | verbose | validate >]

example:

RouterE> **ping dipaddress pktsize 150**

Ping Symbols

Symbol	Meaning
!	Receipt of ping echo reply
.	No reply
U	Got reply with destination unreachable
I	User pressed **Ctrl + C**
&	Got reply with ICMP type “Time Exceeded”
D	Got reply from different hosts
?	Any other ICMP type. To get more info try ping x.x.x.x verbose

applicable models:

All models.

15 SAVE

Use the **save** or **write** commands to save system configurations to either flash memory or a network host.

The **save** or **write** commands are as follows:

Save Commands
write local
write memory
write network
write terminal
save local
save network

save local

This command saves a system configuration to flash memory. Use this command each time the system configuration changes and after creating command aliases. This allows the system to boot from the latest configuration upon subsequent power-up or reboot. Optionally, assign a file name to the saved configuration; if a file name is not specified, the default file (system.cfg) is used.

A message confirming a successful save appears after the command is executed.

parameter	definition
file	Name of saved file

syntax:

save local file < *name* >]

example:

Router> **save local file d011899.cfg**

related commands:

write memory
save network

applicable models:

All models.

save network

Saves a system configuration to a network host.

You must specify a file name and the path name to the destination. A message confirming a successful save appears after the command is executed.

parameter	definition
IP address	IP address of network host
file name	Name of saved file, preceded by its path name

syntax:

save network < *IP address*> < *file name* >

example:

Router> **save network 10.1.100.16 /maindir/network01**

related commands:

write memory

applicable models:

All models.

NOTE: The save network command uses tftp. Be sure that a destination file exists and you have proper permission.

write memory

This command saves a system configuration to flash memory. This command is equivalent to the **save local** command.

You should do this each time you change the system configuration and after creating command aliases. This allows the system to boot from the latest configuration upon subsequent power-up or reboot. You may also assign a file name to the saved configuration; if a file name is not specified, the default file (system.cfg) is used.

A message confirming a successful save appears after the command is executed.

syntax:

file < *name* >]

parameter	definition
file	Name of saved local file system memory.

example:

Router> **write memory file d011899.cfg**

related commands:

save local
save network

applicable models:

All models.

write local

This command saves a system configuration to flash memory. Use this command each time the system configuration changes and after creating command aliases. This allows the system to boot from the latest configuration upon subsequent power-up or reboot. Optionally, assign a file name to the saved configuration; if a file name is not specified, the default file (system.cfg) is used.

A message confirming a successful save appears after the command is executed.

parameter	definition
file	Name of saved file

syntax:

write local file < *name* >]

example:

Router> **write local file d011899.cfg**

related commands:

write memory
save network
save local

applicable models:

All models.

write network

Saves a system configuration to a network host.

parameter	definition
event_logs	Upload event log from NCM to network.
flash_file	Upload a flash file from NCM to network.

syntax:

write network < *event_logs*> <flash_file >

example:

Router> **save network 10.1.100.16 /maindir/network01**

related commands:

configure network
save network

applicable models:

All models.

write terminal

Displays the current configuration on the local host. This command is the same as **show configuration running**.

syntax:

write terminal

example:

Router> **write terminal**

related commands:

cshow configuration running

applicable models:

All models.

16 RELOAD

Use the **reload** command to boot the Router system from the current NCM and IC boot parameters. This command disconnects all users from the system and momentarily disrupts traffic through the system.

The **reload** command is as follows:

reload Command

reload

reload

This command reboots the Router system from the current NCM and IC boot parameters.

Use this command to disconnect all users from the system and momentarily disrupts traffic through the system.

To see the current boot parameters of the NCM and IC modules before rebooting, use the **display boot_params** command.

syntax:

reload

example:

Router> **reload**

related commands:

display boot_params

applicable models:

All models.

NOTE: The system will caution you to save the current configuration before rebooting.

17 TELNET

Use the **telnet** command to telnet directly from one Router system to another Router system (or network host). This feature eases network configuration by eliminating the need to exit the Router CLI before establishing a telnet session.

The **telnet** command is as follows:

Telnet Command

telnet

telnet

This command establishes a Telnet session directly from a Router system to another Router system or network host.

parameter	definition
ipaddress	IP address of another Router system or network host
portno	Port number The default is 23.

syntax:

telnet ipaddress < *IP address* > [portno < n >]

example:

Router> **telnet 10.1.100.16**

applicable models:

All models.

18 TEST

Use the **test** commands to conduct tests on WAN interfaces. You may also test the system fans and temperature sensors.

The first-level **test** commands are as follows:

Test Commands
test ct3
test e1
test fan
test t1
test t3
test ussi

test e1

This command accesses next-level commands for conducting tests on an E1 interface.

parameter	definition
e1_no	E1 link(s) to test. 'The range is 1 - 16, depending upon the system.

syntax:

e1 e1_no < n >

example:

RouterE/test> **e1 1**

next-level commands

test e1 bert
test e1 loopback
test e1 monitor_port

applicable models:

All models.

test e1 bert

This command performs a bit error rate test (BERT) on an E1 link.

Before conducting a BERT test, activate an E1 line loopback at a remote system; the remote system will loop the the BERT pattern back to the Router system for bit error rate calculation. The bit error rate is an indicator of overall E1 quality. To activate a remote E1 loopback, use the **test e1 loopback remote line** command.

You may choose the type of BERT pattern to be sent, and you may set the BERT duration. You may abort a BERT test at any time by typing **no bert n** at the test/e1> prompt, where **n** is the E1 on which the BERT was conducted.

You may apply a BERT pattern to multiple E1s by using a dash to indicate a range or a comma to separate individual E1s.

parameter	definition
0s	All-zeros pattern.
1s	All-ones pattern.
2^15	Pseudorandom pattern with no more than 14 consecutive zeros and no more than 15 consecutive ones. Use this pattern for testing at data rates above 19.2 kbps (G.703-standard test).
2^20	A pattern of no more than 19 consecutive zeros and no more than 20 consecutive ones.
2^23	Pseudorandom signal with no more than 22 consecutive zeros and no more than 23 consecutive ones. This pattern provides the highest stress of all BERT patterns.
1in3	A pattern of no more than 2 consecutive zeros and a one.
1in7	A pattern of no more than 6 consecutive zeros and a one.
QRW	Quasi-random waveform pattern (default). This is pattern simulates live data.
interval	Duration for BERT test. The range is 1-1092 minutes; the default is continuous.

syntax:

[no] bert < 0s | 1s | 2^15 | 2^20 | 2^23 | 1in3 | 1in7 | QRW > interval < n >]

example:

RouterE/test/e1 3> **bert interval 5**

The example above sends a BERT pattern to the remote system on the currently selected E1 link. The BERT pattern (a QRW signal) is sent for 5 minutes. To view the BERT error count during the test, use the **display module test e1** command.

related commands:

test e1 loopback remote line
display module test e1

applicable models:

All models.

test e1 loopback

This command accesses next-level commands for activating and deactivating loopbacks on a E1 link.

syntax:

loopback

example:

RouterE/test/e1 3> **loopback**

next-level commands

test e1 loopback inward1
test e1 loopback inward1_analog
test e1 loopback inward2
test e1 loopback line
test e1 loopback payload

applicable models:

All models.

test e1 loopback inward1

This command configures an E1 interface for local inward 1 loopback.

The inward 1 loopback allows the signal going out to the network to be looped back into the received signal. The loopback takes effect close to the network interface section (LIU), so that the loopback path includes as much of the interface as possible.

syntax:

loopback inward1

example:

RouterE/test/e1 1> **loopback inward1**

related commands:

test e1 loopback inward1_analog
test e1 loopback inward2
test e1 loopback line
test e1 loopback payload

applicable models:

All models.

test e1 loopback inward1_analog

This command configures an E1 interface for local inward 1 analog loopback.

The analog loopback exercises the maximum number of functional blocks. It loops back the actual analog transmit signal back to the receive.

syntax:

loopback inward1_analog

example:

RouterE/test/e1 1> **loopback inward1_analog**

related commands:

test e1 loopback inward1
test e1 loopback inward2
test e1 loopback line
test e1 loopback payload

applicable models:

All models.

test e1 loopback inward2

This command configures an E1 interface for local inward 2 loopback.

The inward 2 loopback also loops back the signal going out to the network back into the received signal, however, this loopback takes effect at the Framer.

syntax:
loopback inward2

example:

RouterE/test/e1 1> **loopback inward2**

related commands:

test e1 loopback inward1
test e1 loopback inward1_analog
test e1 loopback line
test e1 loopback payload

All models.

test e1 loopback line

This command activates an E1 line loopback.
This loopback sends an incoming E1 signal back to a remote system for line testing.

syntax:

[no] loopback line

example:

RouterE/test/e1 3> **loopback line**

To verify loopback status during the test, use the **display module test e1** command.

related commands:

test e1 loopback inward1
test e1 loopback inward1_analog
test e1 loopback inward2
test e1 loopback payload

applicable models:

All models.

test e1 loopback payload

This command activates a local E1 payload loopback.

The loopback sends the incoming E1 back to the remote system. The payload loopback corrects bipolar violations, CRC errors, and frame bit errors before returning the signal.

syntax:

[no] loopback payload

example:

RouterE/test/e1 3> **loopback payload**

To verify loopback status during the test, use the **display module test e1** command.

related commands:

test e1 loopback inward1
test e1 loopback inward1_analog
test e1 loopback inward2
test e1 loopback line

applicable models:

All models.

test e1 monitor_port

This command routes an E1 link to the front panel monitor port.

You may then connect a test set to the port, apply a test signal, and monitor the incoming E1.

parameter	definition
inject	Simultaneously applies a test signal from an external test set to the E1 link, and monitors the incoming E1 signal.
no_inject	Monitors an incoming E1 signal, but not apply a test signal (default).

syntax:

[no] monitor_port [< inject | no_inject >]

example:

RouterE/test/e1 3> **monitor_port inject**

The example above routes the E1 link to the monitor port and injects an external test signal.

applicable models:

All models except OmniAccess 604E.

test fan

This command accesses next-level commands for fan testing and status verification.

syntax:

fan

example:

Router/test> **fan**

next-level commands

test fan run
test fan status

applicable models:

All models except OmniAccess 604.

test fan run

This command turns individual cooling fans on and off.

parameter	definition
fan number	System fan to be turned off or on (1 or 2).

syntax:

[no] fan run < 1 | 2 >

example:

Router/test/fan> **run 1**

The example above turns fan 1 on.

related commands:

test fan status

applicable models:

All models except OmniAccess 604.

NOTE: Only the system administrator (level 1 access) can perform this test.

test fan status

This command displays the status of the fans. If a fan is turned off or has failed, this will be reported.

syntax:

status

example:

Router/test/fan> **status**

related commands:

test fan run

applicable models:

All models except OmniAccess 604.

test t1

This command accesses next-level commands for conducting tests on a T1 interface.

parameter	definition
t1_no	T1 link(s) to test The range is 1 - 16, depending on the Router system.

syntax:

t1 t1_no < n >

example:

Router/test> **t1 1**

next-level commands

test t1 bert
test t1 loopback
test t1 monitor_port

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

test t1 bert

This command performs a bit error rate test (BERT) on a T1 link.

Before conducting a BERT test, activate a T1 line loopback at a remote system; the remote system will loop the BERT pattern back to the Router system for bit error rate calculation. The bit error rate is an indicator of overall T1 quality. To activate a remote T1 loopback, use the **test t1 loopback remote line** command.

You may choose the type of BERT pattern to be sent, and you may set the BERT duration. You may abort a BERT test at any time by typing **no bert n** at the test/t1> prompt, where **n** is the T1 on which the BERT was conducted.

You may apply a BERT pattern to multiple T1s by using a dash to indicate a range or a comma to separate individual T1s.

parameter	definition
0s	All-zeros pattern
1s	All-ones pattern
2^15	Pseudorandom pattern with no more than 14 consecutive zeros and no more than 15 consecutive ones
	Use this pattern for testing at data rates above 19.2 kbps (G.703-standard test).
2^23	Pseudorandom signal with no more than 22 consecutive zeros and no more than 23 consecutive ones
	This pattern provides the highest stress of all BERT patterns.
QRW	Quasi-random waveform pattern (default)
	This is pattern simulates live data.
interval	Duration for BERT test
	The range is 1 - 1092 minutes; the default is continuous.

syntax:

[no] bert < 0s | 1s | 2^15 | 2^23 | QRW > [interval < n >]

example:

Router/test/t1 1> **bert interval 5**

The example above sends a BERT pattern to the remote system on the currently selected T1 link. The BERT pattern (a QRW signal) is sent for 5 minutes. To view the BERT error count during the test, use the **display module test t1** command.

related commands:

test t1 loopback remote line
display module test t1

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

test t1 loopback

This command accesses next-level commands for activating and deactivating loopbacks on a T1 link.

syntax:

loopback

example:

Router/test/t1 1> **loopback**

next-level commands

test t1 loopback line
test t1 loopback payload
test t1 loopback remote

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

test t1 loopback line

This command activates a T1 line loopback.

This loopback sends an incoming T1 signal back to a remote system for line testing.

You may activate a T1 loopback on the Router system by sending an inband or FDL loop-up code from the remote system.

If the T1 link is D4-framed, you must send an inband loop-up code. If it is ESF-framed, you may send an AT&T, or ANSI loop-up code over the FDL.

syntax:

[no] loopback line

example:

Router/test/t1 1> **loopback line**

To verify loopback status during the test, use the **display module test t1** command.

related commands:

display module test t1

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

test t1 loopback payload

This command activates a local T1 payload loopback.

The loopback sends the incoming T1 back to the remote system. The payload loopback corrects bipolar violations, CRC errors, and frame bit errors before returning the signal.

You may activate a T1 loopback on the Router system by sending an inband or FDL loop-up code from the remote system.

If the T1 link is D4-framed, you must send an inband loop-up code. If it is ESF-framed, you may send an AT&T, or ANSI loop-up code over the FDL.

syntax:

[no] loopback payload

example:

Router/test/t1 1> **loopback payload**

To verify loopback status during the test, use the **display module test t1** command.

related commands:

display module test t1

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

NOTE: Payload loopback works only on ESF-framed T1 lines.

test t1 loopback remote

This command accesses next-level commands for activating T1 loopbacks at remote systems.

syntax:

loopback remote

example:

Router/test/t1 1> **loopback remote**

next-level commands

test t1 loopback remote line
test t1 loopback remote payload
test t1 loopback remote smartjack

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

test t1 loopback remote line

This command activates a T1 line loopback at a remote system.

When you enter this command, the Router system sends a FDL loop-up code to the remote system. Upon detecting the loop-up code, the remote system sends the T1 signal back to the Router system. The looped signal path allows you to conduct line testing.

When entering this command, you may specify the type of loop-up code sent to the remote system. The code type will depend on the codes recognized by the remote system.

parameter	definition
ansi_fdl	ANSI T1.403 loop-up code, via the FDL, for ESF-framed T1s (default).
inband_standard	Inband loop-up code for ESF-framed T1s.
inband_alternate	Inband loop-up code for D4-framed T1s.

syntax:

[no] loopback remote line [< ansi_fdl | att_fdl | inband_standard | inband_alternate >]

example:

Router/test/t1 1> **loopback remote line inband_standard**

The example above loops the T1 link at the remote system using an inband ESF standard loop-up code.

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

test t1 loopback remote payload

This command activates a T1 payload loopback at the remote system.

When your enter this command, the Router system sends an FDL payload loop-up code to the remote system.

Upon detecting the loop-up code, the remote system removes BPVs, CRC errors, and frame bit errors before returning the signal to the Router system. The looped signal path allows you to conduct line testing.

parameter	definition
ansi_fdl	ANSI T1.403 loop-up code, via the T1 FDL, for ESF framed T1s (default).
att_fdl	AT&T TR-54016 loop-up code, via the T1 FDL

syntax:

[no] loopback remote payload [< ansi_fdl | att_fdl >]

example:

Router/test/t1 1> **loopback remote payload att_fdl**

The example above loops the T1 link at the remote system by sending an AT&T FDL loop-up code.

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

NOTE: Payload loopback works only on ESF-framed T1 lines.

test t1 loopback remote smartjack

This command configures a remote smartjack interface for remote payload loopback.

parameter	definition
loopback_type	Loopcode for remote smartjack loopback
smart_jack	Smart jack loop-up loopcode (default).

syntax:

[no] smartjack [loopback_type < smart_jack >]

example:

Router/test/t1 1> **smartjack smart_jack**

applicable models:

All models.

test t1 monitor_port

This command routes a T1 link to the front panel monitor port.

You may then connect a test set to the port, apply a test signal, and monitor the incoming T1.

parameter	definition
inject	Simultaneously applies a test signal from an external test set to the T1 link, and monitors the incoming T1 signal.
no_inject	Monitors an incoming T1 signal, but not apply a test signal The default is no_inject.

syntax:

[no] monitor_port [< inject | no_inject >]

example:

Router/test/t1 1> **monitor_port inject**

The example above routes the T1 link to the monitor port and injects an external test signal.

applicable models:

OmniAccess 601, OmniAccess 602, OmniAccess 604

19 TRACE

Use the **trace** command to trace packet routes to destination IP addresses or host names.

The **trace** command is as follows:

Trace Command

trace

trace

Traces packet routes to destination IP addresses or host names.

parameter	definition
dipaddress	IP address or hostname of a remote system.
sipaddress	Source IP address for the probe packet.
protocol	Probe packet protocol type which can be ICMP or UDP.
timeout	Timeout of the probe packet The range is 1 - 3600; the default is 5.
probecnt	Number of probe packets to be sent. The range is 1 - 65535; the default is 3.
minttl	TTL value for the first probe The range is 1 - 255; the default is 1.
maxttl	Maximum value for the TTL The range is 1 - 255; the default is 5.
portno	Destination port used by the UDP probe The range is 1 - 65535; the default is 33434.
symdisp	Toggles the symbolic display on and off. The default is on.

syntax:

trace dipaddress < *ip address* > [*sipaddress* < *ip address* >] [protocol] [timeout < n >] [probecnt < n >] [minttl < n >] [maxttl < n >] [portno < n >] [symdisp < on | off >]

example:

Router> **trace dipaddress 10.1.2.1 timeout 10 probecnt 4 maxttl 6**

applicable models:

All models.

20
FIREWALL COMMANDS

Use the firewall commands to configure protection and management services for the Router router. The firewall commands allow you to:

- Create filters for FTP, HTTP, SMTP, RPC and other traffic
- Protect the network against intrusion attempts including the broad spectrum of Denial of Service attacks such as SYN attacks, Win-nuke attacks, and IP sequence number spoofing
- Manage traffic by creating policies to control connections, bandwidth and the like.

The limits on the number of VPN connections through the firewall are:

- Static VPN connections—5,000
- Dynamic VPN Connections—30,000
- Static Web Alg Connections—10,000
- Static FTP Cntrl Connections—1,000
- Static FTP data Connections—10,000
- Static VPN Selectors—30,000

firewall

Enters the configure mode for all firewall global and firewall map-related commands.

syntax:

firewall map-name

parameter	definition
map-name	A label for map-name mode commands.

example:

To configure firewall global commands, enter:

```
Router/configure>  firewall global
Router/configure/firewall global>
```

To configure corporate firewall map-related commands, enter:

```
Router/configure>  firewall corp
Router/configure/firewall corp>
```

applicable models:

All models.

firewall dos-protect

Enters dos-protect mode, providing access to Denial of Service (DoS) commands.

syntax:

dos-protect

example:

```
Router/configure/firewall global> dos-protect

Router/configure/firewall global/dos-protect>
```

applicable models:

All models.

firewall dos-protect dns-replay-attack

A DNS replay attack occurs when an individual intercepts traffic, analyzes the captured packets and obtains authentication information. They can then use this information to gain access to other systems by reinserting the authenticated packets on the Internet and replaying them.

To disable the DNS replay attack check, use the **no** form of the command. By default it is disabled.

When this command is enabled, the DNS connection limit is 2,000.

syntax:

[no] dns-replay-attack

example:

```
Router/configure/firewall global/dos-protect> dns-replay-attack
Router/configure/firewall global/dos-protect>
```

applicable models:

All models.

dos-protect source-routing

Provides control for source routing checks. After enabling source routing check, the firewall filters out all the datagrams with the strict or loose source routing option enabled. This command is disabled by default. Use the **no** form of the command to disable source routing attack checks.

syntax:

[no] source-routing

example:

```
Router/configure/firewall global/dos-protect>  source-routing
```

applicable models:

All models.

dos-protect syn-flooding

Protects the router from syn-flooding attacks or provides the control for SYN flooding attack checks. To disable the SYN flooding attack check, use the **no** form of the command. By default it is enabled.

syntax:

[no] syn-flooding

example:

```
Router/configure/firewall global/dos-protect>  syn-flooding
```

applicable models:

All models.

dos-protect win-nuke

Provides the control for winnuke attack check. The winnuke attack sends OOB (Out-of-Band) data to an IP address of a Windows machine connected to a network and/or Internet. To disable the win nuke attack check, use the **no** form of the command. This command is disabled by default.

syntax:

[no] *win-nuke*

example:

To enable the winnuke attach check, enter:

```
Router/configure/firewall global/dos-protect>  win-nuke
```

applicable models:

All models.

dos-protect icmp-error

Provides the control for icmp-error attack check. The icmp-error attacks target ICMP (Internet Control Message Protocol) error reporting system. By constructing packets that generate ICMP error responses, an attacker can overwhelm a server's incoming network and cause the server to overwhelm its outgoing network with ICMP responses. To disable the icmp-error attack check, use the **no** form of the command. By default it is enabled.

syntax:

[no] icmp-error

example:

To enable an ICMP-error attack check, enter.

```
Router/configure/firewall global/dos-protect>  icmp-error
```

applicable models:

All models.

dos-protect ftp-bounce

Provides the control for ftp-bounce attack check. In a bounce attack, the hacker uploads a file to the FTP (File Transfer Protocol) server and then requests this file to be sent to an internal server. The file can contain malicious software that destroys data, or it can contain a simple script that executes instructions on the internal server that uses up all the memory and CPU resources. To disable the ftp bounce attack check, use the **no** form of this command. This command is disabled by default.

syntax:

[no] ftp-bounce

example:

To protect against FTP-bounce attacks, enter:

```
Router/configure/firewall global/dos-protect>  ftp-bounce
```

applicable models:

All models.

dos-protect mime-flood

Provides the control for MIME (Multipurpose Internet Mail Extensions) flood attack check. This type of attack is possible on web server. Here the attacker keeps sending numerous request headers of extremely long lengths to the target web server. Over time (and with enough headers), remote attackers can crash the web server or consume massive CPU resources, memory and so on. To disable the MIME flood attack check, use the **no** form of this command. This command is disabled by default

syntax:

[no] mime-flood [header-len] [headers]

parameter	definition
header-len	The MIME header length. Valid length is 256 to 34464 bits (default: 8192). bytes
headers	The number of MIME headers. Valid number is 12—34464 (default 16).

example:

To configure the **mime-flood** attack check with default value, enter:

```
Router/configure/firewall global/dos-protect>  mime-flood
```

To configure the **mime-flood** attack check with 200 headers, with each header 1000 bits long enter:

```
Router/configure/firewall global/dos-protect>  mime-flood
header-len 1000 headers 200
```

applicable models:

All models.

dos-protect ip-unaligned-timestamp

Provides support for an unaligned IP timestamp check. Some operating systems crash if they receive a frame with the IP timestamp option not aligned on a 32-bit boundary. To disable IP unaligned timestamp checks, use the **no** form of this command. This command is disabled by default

syntax:

[no] ip-unaligned-timestamp

example:

To enable IP unaliged timestamp checks, enter:

```
Router/configure/firewall global/dos-protect>  ip-unaligned-timestamp
```

applicable models:

All models.

dos-protect tcp-seq-number-predict

Prevents attempts to predict IP sequence numbers. If an attacker can predict the initial sequence number in the TCP (Transport Control Protocol) handshake, the attacker may be able to hijack the TCP session. This option randomizes the TCP ISNs (Initial Sequence Number) going through the firewall. To disable the tcp-seq-number predict check, use the **no** form of this command. This command is disabled by default.

syntax:

[no] tcp-seq-number-predict

example:

To randomize TCP handshake ISNs, enter:

```
Router/configure/firewall global/dos-protect>
tcp-seq-number-predict
```

applicable models:

All models.

dos-protect tcp-seq-number-range

Provides the control to check TCP sequence number out of range. Attacker can attempt to replay a captured packet through the firewall by brut-force and thus consume the bandwidth as well as the resources of the target CPU. With this check turned on, the firewall allows only those packets that have sequence numbers in a configured range from the last acknowledgement seen on the connection. The range can be configured with value between 20000 and 2147483647. To disable TCP sequence number range check, use the **no** form of this command. By default it is disabled.

syntax:

[no] tcp-*seq-number-range* <number>

parameter	definition
tcp-seq-number-range	The sequence number range. Valid range is 20000—2147483647. The default: is 20000.

example:

To configure the TCP sequence number range to be 21000, enter:

```
Router/configure/firewall global/dos-protect> tcp-seq-number-range
21000
```

applicable models:

All models.

dos-protect enable-all

Enables/disables all the DoS attack checks. To disable all DoS attack checks, use the **no** form of this command.

syntax:

[no] enable-all

parameter	definition
global	

example:

To enable all of the Denial of Service protections, enter:

```
Router/configure/firewall global/dos-protect>  enable-all
```

applicable models:

All models.

firewall logging

Enters logging and provides access to logging related commands.

syntax:

logging

example:

To access the logging mode, enter:

```
Router/configure/firewall global>  logging
Router/configure/firewall global/logging>
```

applicable models:

All models.

logging vpn

Controls the recording of VPN logs. When the VPN log reaches the configured threshold value (default is 100 events), a log is created. By default, logged information is written to console. If a syslog server is configured, the log information is directed to the syslog server and the console. Use the **no** form of the command to set VPN logging to its default value.

syntax:

[no] vpn <count>

parameter	definition
count	The threshold for VPN logging. The range is 1—2147483647, the default is 100 packets.

example:

To configure the router to generate a log message for 10 VPN events, enter:

```
Router/configure/firewall global/logging>  vpn 10
```

applicable models:

All models.

logging attacks

Configures a threshold for logging attacks. Whenever the number of attacks reaches the configured threshold value, a log message is generated. By default, logged information is written to console. If a syslog server is configured, the log information is directed to the syslog server and the console. Use the **no** form of the command to set attack logging to its default value (1).

syntax:

[no] attacks <count>

parameter	definition
limit	The threshold for attack logging. The range is 1—2147483647, the default is 1 packet.

example:

To configure the router to generate a log message for every 25 attacks, enter:

```
Router/configure/firewall global/logging>  attacks 25
```

applicable models:

All models.

logging policy

Configures a threshold for policy based logging. Defines the number of events required against an access policy, to generate a log message for that policy. By default, logged information is written to console. If a syslog server is configured, the log information is directed to the syslog server and the console. Use the no form of the command to reset to default values. By default, the threshold is configured as 1.

syntax:

[no] policy <count>

parameter	definition
limit	The threshold for policy logging. The range is 1—2147483647, the default is 1 packet.

example:

To configure the router to generate a log message for every 20 events.

```
Router/configure/firewall global/logging>  policy 20
```

applicable models:

All models.

firewall ip-reassembly

Enters the ip-reassembly mode and provides access to IP-reassembly-related commands.

syntax:

ip-reassembly

example:

To access IP-reassembly-related commands, enter:

```
Router/configure/firewall global>  ip-reassembly
Router/configure/firewall global/ip-reassembly>
```

applicable models:

All models.

ip-reassembly enable

Enables IP packet reassembly. To disable IP reassembly, use the no form of the command. By default, IP reassembly is enabled.

IP allows packets to be split in transit and reassembled on delivery. This allows longer packets to be routed through intermediate networks that have rules limiting packets to lengths smaller than the routed packet. The oversized packet can be broken up into pieces small enough to pass. The IP reassembly feature allow the split packets to be reassembled upon delivery.

syntax:

[no] enable

example:

To enable IP reassembly, enter:

```
Router/configure/firewall global/ip-reassembly>  enable
```

applicable models:

All models.

ip-reassembly packet-size

Sets the maximum size of the IP packet. Use the **no** form of the command to set packet size to its default value.

syntax:

packet-size <size>

parameter	definition
size	The size in bits of the packet. The range is 1—65535. The default is 65535.

example:

To set the IP packet size to 10000 bits, enter:

```
Router/configure/firewall global/ip-reassembly>  packet-size 10000
```

applicable models:

All models.

ip-reassembly fragment-size

Sets the maximum allowable fragment size of the IP packet. Use the **no** form of the command to set the fragment size to its default value.

syntax:

fragment-size <size>

parameter	definition
size	The length of the fragment header. The range is 1—65535. The default is 28 bits.

example:

To set the fragment header to 5000 bits, enter:

```
Router/configure/firewall global/ip-reassembly>  fragment-size 5000
```

applicable models:

All models.

ip-reassembly fragment-count

Sets the maximum number of fragments allowed per IP packet. This value limits the number of fragments into which a packet can be fragmented. Use the **no** form of the command to set fragment count to its default value.

syntax:

fragment-count <count>

parameter	definition
count	The number of fragments. The range is 1—214748364. The default is 44.

example:

To limit the number of fragments to 5000, enter:

```
Router/configure/firewall global/ip-reassembly>  fragment-count
5000
```

applicable models:

All models.

ip-reassembly timeout

Sets the IP reassembly timeout value. If a fragmented packet is not reassembled within this time limit, the packet is discarded. Use the **no** form of the command to set the timeout to its default value.

syntax:

timeout <time>

parameter	definition
time	A packet has to be reassembled before this amount of time expires. The valid range is 11—120 seconds, and the default is 60.

example:

To allow fragmented packets up to 65 seconds to be reassembled (after which they are dropped), enter:

```
Router/configure/firewall global/ip-reassembly>  timeout 65
```

applicable models:

All models.

firewall timeout

Enters the configure level for firewall timeout commands.

syntax:

timeout

example:

To enter the timeout mode for firewalls, enter:

```
Router/configure/firewall global>  timeout
Router/configure/firewall global/timeout>
```

applicable models:

All models.

timeout general

Modifies the default timeout values for protocols like TCP, UDP, ICMP, FTP and DNS. Use the **no** form of the command to restore timeouts to their default values.

syntax:

[no] general <**tcp| udp| icmp| tcp-reset| ftp-inactivity| dns-inactivity**> <time>

parameter	definition
tcp	Specifies the Transport Control Protocol timeout.
udp	Specifies the User Datagram Protocol timeout.
icmp	Specifies the Internet Control Message Protocol timeout.
tcp-reset	Specifies the Transport Control Protocol reset timeout.
ftp-inactivity	Specifies how long the File Transport Protocol will wait for a response.
dns-inactivity	Specifies how long the Domain Name Service will wait for a response.
time	The timeout value. The valid range is: 0—65535 seconds.

example:

To configure TCP to timeout after waiting 1000 seconds, enter:

```
Router/configure/firewall global/timeout>  general tcp 1000
```

applicable models:

All models.

timeout service

Configures the timeout service record.

syntax:

[no] service <name> | protocol <tcp | udp> <port> <time>

parameter	definition
name	The name of the service.
tcp	Specifies if the packet is TCP.
udp	Specifies if the packet is UDP.
port	Specifies the port number for the service.
time	The timeout value. The range is 0—65535 seconds.

example:

To create a UDP service named Test on port 4444, with a timeout value of 600 seconds, enter:

Router/configure/firewall global/timeout> **service Test udp 4444 600**

applicable models:

All models.

firewall max-connection-limit

Controls the number of connections through the firewall.

syntax:
max-connection-limit <map-name> <connections>

parameter	definition
map-name	Specifies the name of the map.
connections	Specifies the maximum number of connections allowed. The range is 1—29912.

example:

To limit the number of connections through the firewall from **mainmap** to 22, enter:

```
Router/configure/firewall global>  max-connection-limit mainmap 22
```

applicable models:

All models.

firewall object

Enters object mode and provides access to object-related commands like ftp-filter, http-filter etc. The objects will be global objects if created in **firewall global mode** or map specific if created when in a specific map mode.

syntax:

object

example:

To create a global object, enter:

```
Router/configure/firewall global>  object
Router/configure/firewall global/object>
```

To create an object specific to the `corp` map, enter

```
Router/configure/firewall corp>  object
Router/configure/firewall corp/object>
```

applicable models:

All models.

firewall port-trigger

Configures a new trigger record. Port triggering lets you define an application specific customized firewall policy, which lets you enable configured same-direction or reverse-direction conduits through the firewall, when the trigger application is launched.

Trigger application is defined by the trigger IP address, protocol and port. Here conduit is referred as combination of transport-protocol and port number. Use the **no** form of the command to remove the existing records or to remove the entries from forward-direction or reverse direction.

syntax:

[no] port-trigger <trigger-name> [protocol <tcp | udp>] [port<1-65535>] [address <ipaddress | any>] [forward-direction <tcp | udp> <start-port> <end-port>] [reverse-direction <tcp | udp> <start-port> <end-port>] [timeout <1-600-2147483647] [enable | disable]

parameter	definition
trigger- name	The name string of the trigger.
protocol	Specifies if the protocol is TCP or UDP.
address	IP address of the trigger.
forward-direction	Specifies the protocol (tcp or udp) and the port numbers (start and end port) to open in the same direction as the control connection established
reverse-direction	Specifies the protocol (tcp or udp) and the port numbers (start and end ports) to open in the opposite direction as the control connection established.
timeout	The timeout value. The range is 0—65535 seconds.
enable/disable	Enables or disables same-direction or reverse-direction conduits through the firewall.

example:

To create a new trigger record for an application running on 10.1.1.1 using TCP with port number 3001, which opens up UDP ports 6000 to 6020 in the reverse direction, enter:

```
Router/configure/firewall global> trigger app1 proto tcp port 3001
address 10.1.1.1 reverse-direction udp 6000 6020
```

firewall stealth-mode

Stops the firewall sending TCP reset packets when there is no corresponding matching policy for an incoming packet. This command is applicable for both firewall global and firewall map.

By default, this feature is disabled.

syntax:

[no] stealth-mode <map-name>

parameter	definition
map-name	The name of the map.

example:

To enable stealth mode for the map **global**, enter:

```
Router/configure/firewall global>  stealth-mode global
```

To disable stealth mode for the map **global**, enter:

```
Router/configure/firewall global>  no stealth-mode global
```

To enable the stealth mode on corp. Any one of the following commands can be executed.

```
Router/configure/firewall corp> stealth-mode corp
```

or

```
Router/configure/firewall global> stealth-mode corp
```

or

```
Router/configure/firewall corp> stealth-mode
```

To disable the stealth mode on corp. Any one of the following commands can be executed.

```
Router/configure/firewall corp> no stealth-mode corp
```

or

```
Router/configure/firewall corp> no stealth-mode
```

or

```
Router/configure/firewall global> no stealth-mode corp
```

applicable models:

All models.

firewall url-key-filter

Filters web access for out bound connections, based on the key words in URLs. Use the **no** form of the command to undo the filtering effect for any key word. Users can filter up to 20 key words at a time.

syntax:

[no] url-key-filter <key names>

parameter	definition
key names	The list of word strings, separated by a space. The maximum list size is 20.

example:

To prevent users from connecting to websites with the URL key names **games** and **movies** in them, enter:

```
Router/configure/firewall global >  url-key-filter games movies
```

To disable URL key filtering temporarily, enter:

```
Router/configure/firewall global >  no url-key-filter
```

To resume URL key filtering, enter:

```
Router/configure/firewall global >  url-key-filter
```

To allow users to access websites containing the URL key name **games**, enter:

```
Router/configure/firewall global >  no url-key-filter games
```

applicable models:

All models.

object ftp-filter

Creates an FTP application filter object. Use the **no** form of the command to remove the existing ftp-filter object or remove the FTP commands from an existing ftp-filter object. This application filter object can be applied to any number of policies in any map if created in firewall global mode. If created in a specific map, then this application filter can be applied to any number of policies, but only for this specific map.

syntax:

[no] ftp-filter object-name [permit] [deny] [log]

parameter	definition
object-name	The name of the FTP-filter object.
permit	The list of FTP commands to allow. Includes: put, get, ls, mkdir, cd, and pasv. Enter commands listed as strings separated by a space.
deny	The list of FTP commands to prevent. Includes: put, get, ls, mkdir, cd, and pasv. Enter commands listed as strings separated by a space.
log	Enables or disables logging.

example:

To create a global FTP application filter named myftp that permits put, get, and ls commands, enter:

```
Router/configure> firewall global
Router/configure/firewall global> object
Router/configure/firewall global/object>  ftp-filter myftp permit
put get ls
```

To remove the FTP commands **put** and **get** from a global FTP permit application filter named **myftp**, enter:

```
Router/configure/firewall global/object>  no ftp-filter myftp
permit put get
```

To create a global FTP application filter named **ftpdeny** that prevents users from changing directory levels or making new directories, enter:

```
Router/configure/firewall global/object>  ftp-filter ftpdeny deny
mkdir cd
```

To delete a global FTP application filter named **myftp**, enter:

```
Router/configure/firewall global/object>  no ftp-filter myftp
```

To create a map-specific FTP application filter named **corpftp** that allows permit, get, and ls commands, enter:

```
Router/configure> firewall corp
Router/configure/firewall corp>  object
Router/configure/firewall corp/object>  ftp-filter corpftp permit
put get ls
```

applicable models:

All models.

object http-filter

Creates an HTTP (Hyper Text Transfer Protocol) application filter object. Use the **no** form of the command to delete the existing http-filer object or remove the file extensions from an existing http-filter object.

This application filter object can be applied to any number of policies if created in firewall global mode. If created in a specific map, then this application filter can be applied to any number of policies, but only for this specific map.

syntax:

[no] http-filter object-name [deny] [log]

parameter	definition
object name	The name of the HTTP filter.
deny	Lists web extensions that cannot be executed (for example, java, activex, jar, mar, and *.url). The list is limited to 10 extensions (excluding java, activex, and jar) entered as strings separated by space.
log	Enables or disables logging.

example:

To create a global HTTP application filter named **myhttp** that permits all the URLs except Java and Activex, enter:

```
Router/configure> firewall global
Router/configure/firewall global> object
Router/configure/firewall global/object>  http-filter myhttp deny
java activex
```

To create a global HTTP application filter named **httpdeny** that permits all URLs but filters out Java applications and .gif and .jpg files, enter:

```
Router/configure/firewall global/object>  http-filter httpdeny deny
java *.gif *.jpg
```

To remove HTTP file extensions .gif from a global HTTP application filter named **httpdeny**, enter:

```
Router/configure/firewall global/object>  no http-filter httpdeny
deny *.gif
```

To delete a global HTTP application filter named **httpdeny**, enter:

```
Router/configure/firewall global/object>  no http-filter httpdeny
```

To create a map-specific HTTP application filter named **corphttp** that allows all URLs but filters out Java applications and files with .exe extensions, enter:

```
Router/configure/firewall corp/object>  http-filter corphttp deny
java *.exe
```

applicable models:

All models.

object nat-pool

Configures a NAT (Network Address Translation) pool of IP addresses with specified types. The type can be static, dynamic, or PAT (Port Address Translation). Use the **no** form of this command to delete the existing NAT pool object.

This NAT pool can be applied to any policy in any map, if created under firewall global. If created under a map, then this NAT pool can be applied to any policy, but only in that particular map.NAT pool applied to outbound policies (called forward NAT) and to the inbound policies (called reverse NAT). NAT pool of type PAT can be shared across any number of policies, but sticking to the visibility of the NAT pool.

syntax:

[no] nat-pool object-name <**static| dynamic| pat**> <nat-start-ip> [nat-end-ip]

parameter	definition
object name	The name of the NAT pool.
nat-type	NAT addresses can be: static: Addresses are configured for each device and do not change dynamic: Addresses are assigned from the NAT pool as needed and for a fixed length of time pat: A single address is assigned to all devices, with arbitrary port numbers.)
nat-start-ip	The starting IP address in the NAT pool range.
nat-end-ip	The ending IP address in the NAT pool range.

NOTE: The forward NAT private address are specified in the firewall policy (policy source address), public address are specified in the NAT pool. Reverse NAT private address are specified in the NAT pool, public address are specified in the firewall policy (policy destination address).

NOTE: All IP addresses are in IPv4 format.

example:

To configure a NAT pool of type one-to-one with a source address range (10.1.1.1 to 10.1.1.10) that will be translated to the address range (20.1.1.1 to 20.1.1.10), enter:

```
Router/configure/firewall global/object>  nat-pool staticpool
static 20.1.1.1 20.1.1.10
```

To configure a global NAT pool of type many-to-many with a source address range 10.1.1.20 to 10.1.1.30 that will be translated to the address range 20.1.1.20 to 20.1.1.25, enter:

```
Router/configure/firewall global/object>  nat-pool dynamicpool dynamic
20.1.1.20 20.1.1.25
```

To configure a global NAT pool of type many-to-one (or PAT) with a source address range of 10.1.1.40 to 10.1.1.50 that will be translated to the address 50.1.1.5, enter:

```
Router/configure/firewall global/object>  nat-pool patpool pat 50.1.1.5
```

To delete a global NAT pool of type many-to-one (or PAT), enter:

```
Router/configure/firewall global/object>  no nat-pool patpool pat
50.1.1.5
```

To configure a map-specific NAT pool of type one-to-one with a source address range 10.1.1.1 to 10.1.1.10)that will be translated to the address range 20.1.1.50 to 20.1.1.60, enter:

```
Router/configure> firewall corp
Router/configure/firewall corp>  object
Router/configure/firewall corp/object>  nat-pool corpspool static
20.1.1.50 20.1.1.60
```

applicable models:

All models.

object rpc-filter

Creates a RPC (Remote Procedure Call) application filter object. Use the **no** form of the command to delete the existing RPC-filter object or remove the RPC numbers from an existing RPC-filter object. This application filter object can be applied to any number of policies in any map, if created under firewall global. If created under a map, then this application filter can be applied to any number of policies, but only in that particular map.

syntax:

[no] rpc-filter object-name [permit] [deny] [log]

parameter	definition
object-name	The name of the RPC filter.
permit	Lists RPC numbers that are allowed. Up to 25 numbers can be listed as strings separated by spaces.
deny	Lists RPC numbers that are prohibited. Up to 25 numbers can be listed as strings separated by spaces.
log	Enables or disables logging.

example:

To create a global RPC application filter named **myrpc** to permit only RPC program numbers 1, 10,and 100, enter:

```
Router/configure> firewall global
Router/configure/firewall global>  object
Router/configure/firewall global/object>  rpc-filter myrpc permit 1
10 100
```

To create a global RPC application filter named **rpcdeny** that prevents users from using RPC program numbers 40 and 50, enter:

```
Router/configure/firewall global/object> rpc-filter rpcdeny deny 40
50
```

To remove the RPC program number 10 from a global RPC permit application filter named **myrpc**, enter:

```
Router/configure/firewall global/object>  no rpc-filter myrpc
permit 10
```

To delete the global RPC application filter named **myrpc**, enter:

```
Router/configure/firewall global/object>  no rpc-filter myrpc
```

To create a map-specific RPC application filter that only permits RPC program numbers 1, 10, and 100, enter:

```
Router/configure> firewall corp
Router/configure/firewall corp> object
Router/configure/firewall corp/object>  rpc-filter corprpc permit 1
10 100
```

applicable models:

All models.

object smtp-filter

Creates an SMTP (Simple Mail Transport Protocol) application filter object. Use the **no** form of the command to delete the existing SMTP filter object or remove SMTP commands from an existing SMTP filter object. This application filter object can be applied to any number of policies in any map, if created under firewall global. If created under a map, then this application filter can be applied to any number of policies, but only in that particular map.

syntax:

{no] smtp-filter object-name [permit] [deny] [log]

parameter	definition
object name	The name of the SMTP filter.
permit	Lists SMTP commands that are allowed. Includes: `hello mail rcpt data quit send saml reset vrfy expn` Enter commands as strings separated by a space.
deny	Lists the SMTP commands that are prohibited. Up to ten commands can be listed as strings separated by spaces.
log	Enables or disables logging.

example:

To create a global SMTP application filter named **mysmtp** that allows only specified commands, enter:

```
Router/configure/firewall global/object>  smtp-filter mysmtp permit helo mail data
```

To remove the SMTP command helo from the global SMTP permit application filter named **mysmtp**, enter:

```
Router/configure/firewall global/object>  no smtp-filter mysmtp permit helo
```

To create a global SMTP application filter named **smtpdeny** that prevents users from using commands like vrfy, enter:

```
Router/configure/firewall global/object> smtp-filter smtpdeny deny vrfy
```

To delete a global SMTP application filter named **mysmtp**, enter:

```
Router/configure/firewall global/object> no smtp-filter mysmtp
```

To create a map specific SMTP application filter named **corpsmtp** that allows only specified commands, enter:

```
Router/configure>  firewall corp
Router/configure/firewall corp>  object
Router/configure/firewall corp/object>  smtp-filter corpsmtp permit helo mail data
```

applicable models:

All models.

object schedule

Creates a schedule object. Add up to three entries in a single schedule record. Use the no form of the command to delete the existing objects or to delete individual entries in a specific schedule object. This schedule object can be applied to any number of policies in any map, if created under firewall global. If created under a map then this schedule object can be applied to any number of policies, but only in that particular map. Firewall policies configured with schedule objects are only activated in the time defined in the schedule object.

syntax:

[no] schedule object-name [week-day] [start-time] [end-time]

parameter	definition
object-name	The name of the schedule object.
week-day	The starting and ending day of the week (sun|mon|tue|wed|thu|fri|sat) separated by a space.
start-time	The time to start the schedule on the specified days. Enter the time in hours and minutes separated by a space.
end-time	The time to stop the schedule on the specified days. Enter the time in hours and minutes, separated by a space.

example:

To create a schedule object named **getmail**, starting at 6 AM and ending at 7 PM, Monday through Friday, enter:

```
Router/configure/firewall global/object>  schedule getmail week-day mon fri start-time 6 0 end-time 19 00
```

To add an entry to a global schedule object named **getmail**, starting at 9:00 AM and ending at 2:30 PM on Saturday, enter:

```
Router/configure/firewall global/object>  schedule getmail week-day sat sat start-time 9 0 end-time 14 30
```

To delete an entry from a global schedule object named **getmail**, starting at 9:00 AM and ending at 6:30 PM, Monday through Friday, enter:

```
Router/configure/firewall global/object>  no schedule getmail week-day mon fri start-time 9 0 end-time 18 30
```

To delete a global schedule object named **getmail**, enter:

```
Router/configure/firewall global/object>  no schedule getmail
```

To create a map-specific schedule object named **getmail**, starting at 9:00 AM and ending at 6:30 PM, Monday through Friday, enter:

```
Router/configure> firewall corp
Router/configure/firewall corp>  object
Router/configure/firewall corp/object>  schedule getmail week-day mon
fri start-time 9 0 end-time 18 30
```

applicable models:

All models.

object service

Creates a new service object. Specify either a port number or port number ranges (up to three) in a single service object. Use the **no** form of the command to remove existing service objects or to remove individual port range entries in the specified service object.

This service object can be used in any number of policies in any map, if created under firewall global. If created under a map then this service object can be used in any number of policies, but only in that particular map.

syntax:

[no] service object-name <tcp | udp> [port]

parameter	definition
object-name	The name of the service object.
tcp, udp	Defines the type of transport protocol: TCP or UDP.
port	Specifies a port number or port number range. Enter a port number or the starting and ending port numbers as strings separated by a space.

example:

To create a new service object named **servport** using TCP on port 2001, enter:

```
Router/configure/firewall global/object>  service servport tcp port
2001
```

To create a global service object named **servrange** using UDP from port 1000 to 1005 and 3000 to 3002, enter:

```
Router/configure/firewall global/object>  service servrange udp port
1000 1005
Router/configure/firewall global/object>  service servrange udp port
3000 3002
```

To remove port number range entry from a global service object servrange, enter:

```
Router/configure/firewall global/object>  no service servrange udp port
1000 1005
```

To delete a global service object named **servport**, enter:

```
Router/configure/firewall global/object>  no service servport tcp
```

To create a map-specific service object named **corpserv** using TCP on port 2001, enter:

```
Router/configure> firewall corp
Router/configure/firewall corp>  object
Router/configure/firewall corp/object>  service corpserv tcp port 2001
```

applicable models:

All models.

object address

Creates a new address object. Specify in a single address object either an address with a prefix length or up to three address range entries. Use the **no** form of the command to remove the existing address objects or to remove individual address range entries in the specified address object. This address object can be used in any number of policies in any map, if created under firewall global. If created under a map then this address object can be used in any number of policies, but only in that particular map.

syntax:

[no] address object-name [ipaddress]

parameter	definition
object-name	The name of the address object.
ipaddress	Either the starting and ending IP address in the range, or the IP address and prefix length, separated by a space.

example:

To create a new address object (**singAddr**) for IP address10.1.1.10 32 and prefix length 32, enter:

```
Router/configure/firewall global/object>  address singAddr
10.1.1.10 32
```

To create a new address object **addrange** with a range of addresses from 10.1.1.11 to 10.1.1.15 and 20.1.1.1 to 20.1.1.10, enter:

```
Router/configure/firewall global/object>  address addrange 10.1.1.11
10.1.1.15
Router/configure/firewall global/object>  address addrange 20.1.1.1
20.1.1.10
```

To remove the address range entry from the global address object **addrange**, enter:

```
Router/configure/firewall global/object>  no address addrange 20.1.1.1
20.1.1.10
```

To delete a global address object named **addrange**, enter:

```
Router/configure/firewall global/object>  no address singAddr
```

To create a map-specific address object named **corpaddr** for IP address 10.1.1.10 and prefix length 32, enter:

```
Router/configure>  firewall corp
Router/configure/firewall corp>  object
Router/configure/firewall corp/object>  address singAddr 10.1.1.10 32
```

applicable models:

All models.

firewall interface

Configures one or more interfaces for a map. Up to 32 interfaces are supported, with a maximum of five interfaces at a time. Use the **no** form of the command to remove one or more interfaces from a map.

syntax:

[no] interface interface-list

parameter	definition
interface	The list of interfaces, where interfaces can be ethernet0, ethernet1, interface name, or interface name:>pvc-number. Enter up to five interfaces as strings separated by spaces.

example:

To add the interface Ethernet0 to the map **corp**, enter:

```
Router/configure> firewall corp
Router/configure/firewall corp>  interface ethernet0
```

To add the interfaces wan 1, wan2, and wan3 to the map **internet**, enter:

```
Router/configure> firewall internet
Router/configure/firewall internet>  interface wan1 wan2 wan3
```

To add the frame relay interface **wan4** with pvc number **16** and **wan5** with pvc number **18** to the map **internet**, enter:

```
Router/configure/firewall internet>  interface wan4:16 wan5:18
```

To add the subinterface **ethernet1.1** to the map **dmz**, enter:

```
Router/configure> firewall dmz
Router/configure/firewall dmz>  interface ethernet1.1
```

To remove **wan3** and the frame relay interface **wan4** with pvc number **16** from the map **internet**, enter:

```
Router/configure>  firewall internet
Router/configure/firewall internet>  no interface wan3 wan4:16
```

applicable models:

All models.

firewall policy

Configures a firewall policy for a specific map. Use the **no** form of the command to delete a policy from the map. The maximum number of policies per map is 1024.
While creating a firewall policy, specify the following:
priority can range from 1 to 1024 which is a unique number for any given **map**.
address (if not specified, then it is taken as **any**). User is allowed to specify an ip address with a prefix-length or a range of address or a predefined address object for source and destination.
service or the combination of (**protocol** and **port** numbers) needs to be specified (if nothing specified, then it is taken as any protocol for any source and destination port numbers).
Source and destination ports can be specified as a single **port** number or as a range of port numbers.
Specify PAT address directly for the **nat-ip** parameter while defining the firewall policy. Other modes of NAT can be achieved by creating the nat-pool and later attaching it to the firewall policy.
While configuring a firewall policy for a self-traffic, specify **self** for the parameter **traffic**. By default firewall policy is **transit**.

syntax:

[no] policy priority direction [action] [address] [service] [protocol] [port] [traffic] [user-group] [nat-ip] [port-map] [log]

parameter	definition
priority	The policy priority. Valid range is: 1—1024.
direction	The direction of the policy. Direction can be: out — outgoing direction in—incoming direction
action	The work performed by the policy. Work can be one of two values: permit -- permit rule (the default) deny -- deny rule
address	A list (maximum of four strings separated by spaces) of the following types of addresses: <src-ip> <prefix-len> <dst-ip> <prefix-len> OR <src-start> <src-end> <dst-start> <dst-end> OR <src-object> <dst-object>
service	The service name.
protocol	The protocol type. Can be one of tcp/udp/icmp/ah/esp/gre/any.
port	The port number or port range (a maximum of four strings separated by spaces.) Can be one of: <src-start> <src-end> <dst-start> <dst-end> OR <src-port> <dst-port>
traffic	The type of traffic. Can be one of the following values: transit -- transit traffic (the default) self -- self traffic
user-group	The user group name.
nat-ip	The IP address for PAT.
port-map	Port to Application Mapping - for reverse NAT only.
log	Enables or disables logging. Default is enable.

example:

To add a transit policy to permit FTP from the map **corp**, enter:

```
Router/configure>  firewall corp
Router/configure/firewall corp>  policy 1 out service ftp permit
Router/configure/firewall corp/policy 1 out>
```

To add a transit policy to permit TCP traffic from 10.1.1.1 with prefix-len 24 to any IP address, enter:

```
Router/configure>  firewall corp
Router/configure/firewall corp>  policy 2 out protocol tcp address
10.1.1.1 24 any any permit
Router/configure/firewall corp/policy 2 out>
```

To add a transit policy to deny TCP traffic with address defined in address object **sobj** and **dobj**, enter:

```
Router/configure>  firewall corp
Router/configure/firewall corp>  policy 3 out protocol tcp address sobj
dobj deny
Router/configure/firewall corp/policy 3 out>
```

To add a transit policy to permit TCP traffic from 120.1.1.1 to any with source port number 100 to 200 and destination port number from 300 to 400, enter:

```
Router/configure/firewall corp>  policy 4 out protocol tcp address
120.1.1.1 32 any any port 100 200 300 400 permit
Router/configure/firewall corp/policy 4 out>
```

To delete a transit policy with priority 3, enter:

```
Router/configure/firewall corp>  no policy 3 out
```

To add a self policy to permit TCP traffic from 100.1.1.1 with a prefix-len of 24 from internet map, enter:

```
Router/configure>  firewall internet
Router/configure/firewall internet>  policy 10 out protocol tcp address
10.1.1.1 24 any any permit self
Router/configure/firewall internet/policy 10 out>
```

To enter an existing policy sub-tree, enter

```
Router/configure>  firewall corp
Router/configure/firewall corp>  policy 4 out
Router/configure/firewall corp/policy 4 out>
```

To modify the source and destination address configuration of an existing firewall policy, enter

```
Router/configure/firewall corp> policy 4 out address 21.1.1.1 21.1.1.5
any any permit
Router/configure/firewall corp/policy 4 out>
```

To remove the source and destination address configuration of an existing firewall policy, enter

```
Router/configure/firewall corp> policy 4 out address any any
Router/configure/firewall corp/policy 4 out>
```

To add a inbound firewall policy for a existing user-group test (remote access feature), enter

```
Router/configure/firewall corp>  policy 14 in 13.1.1.1 32 any any
user-group test permit
Router/configure/firewall corp/policy 14 in>
```

To add a firewall policy for out bound traffic from 40.1.1.1 to do PAT with address 90.1.1.1, enter

```
Router/configure/firewall corp>  policy 15 out 40.1.1.1 32 any any
nat-ip 90.1.1.1 permit
Router/configure/firewall corp/policy 15 out>
```

To add an inbound policy for a web application with destination ip address of 70.1.1.6 prefix-len of 32, a nat-ip address of 10.1.1.10, and an port to application mapping (PAM) of 8080, enter:

```
Router/configure/firewall dmz> policy 14 in address any 70.1.1.6 32
service web nat-ip 10.1.1.10 nat-port 8080
```

applicable models:

All models.

policy max-connection-limit

Specifies the maximum number of connections for a given policy at any given time. Use the **no** form of the command to restore the default value (the default value is the maximum number of connections for the current map).

syntax:

[no] max-connection-limit <*max-number-connections*>

parameter	definition
max-number-connections	Specifies the maximum number of connections at any one time. Valid range is 1—29912. The default is the connection limit of the map.

example:

To limit the maximum number of connections to 20, enter:

```
Router/configure/firewall corp/policy 4 out>  max-connection-limit
20
```

applicable models:

All models.

policy connection-rate

Maximum number of connections for a given policy in a particular time. Use the **no** form of this command to disable this feature for this policy (disabled by default). The seconds parameter defaults to 1 second if not specified.

syntax:

[no] connection-rate <number-connections> [seconds]

parameter	definition
number-connections	Specifies the number of connections. Valid range is 1-29912.
seconds	Specifies the time in seconds. Valid range is 1-36000. Default is 1 second.

example:

To limit the maximum number of connections to 2000 for 2 seconds, enter:

```
Router/configure/firewall corp/policy 4 out >  connection-rate 2000
2
```

To disable the connection-rate feature for this policy, enter:

```
Router/configure/firewall corp/policy 4 out >  no connection-rate 2000
2
```

applicable models:

All models.

policy policing

Specifies the maximum number of packets for a given policy per second. Use the **no** form of this command to disable the policing feature for this policy (disabled by default).

syntax:

[no] policing <packets-*per*-second>

parameter	definition
packets-*per*-second	Specifies the maximum number of packets per second. Valid range is 1—2147483647.

example:

To limit the maximum number of packets to 4000 per second, enter:

```
Router/configure/firewall corp/policy 4 out >  policing 4000
```

To disable the policing feature for this policy, enter:

```
Router/configure/firewall corp/policy 4 out >  no policing 4000
```

applicable models:

All models.

policy bandwidth

Specifies the maximum number of kilobytes for a given policy per second. Use the **no** form of this command to disable the bandwidth-limiting feature for this policy (disabled by default).

syntax:

[no] policy bandwidth <kilobytes-*per*-second>

parameter	definition
kilobytes-*per*-second	Specifies the maximum number of kilobytes per second. Valid range is: 1—4194303.

example:

To limit the maximum number of kilobytes to 4 per second, enter:

```
Router/configure/firewall corp/policy 4 out >  bandwidth 4
```

To disable the bandwidth limiting feature for a policy, enter:

```
Router/configure/firewall corp/policy 4 out > no bandwidth 4
```

applicable models:

All models.

policy enable

Disables or enables an existing policy of a specific map based on the requirement. By default the policy is enabled.

syntax:

[no] enable

example:

To disable a firewall policy, enter:

```
Router/configure/firewall corp/policy 4 out >  no enable
```

To enable a firewall policy that was previously disabled, enter:

```
Router/configure/firewall corp/policy 4 out >  enable
```

applicable models:

All models.

policy apply-object

Applies an object for a particular policy in a given map. Use the **no** form of the command to detach an object from the policy.

syntax:

[no] apply-object <object-type> <object-name>

parameter	definition
object-type	The type of object. Can be one of: ftp-filter, http-filter, smtp-filter, rpc-filter, nat-pool, or schedule.
object-name	The name of the object.

example:

To apply the ftp-filter to the policy **myftp**, enter:

```
Router/configure/firewall corp/policy 4 out>  apply-object
ftp-filter myftp
```

applicable models:

All models.

clear firewall statistics

Clears the firewall statistics.

syntax:
clear firewall statistics

example:
To clear the firewall statistics, enter:

```
Router> clear firewall statistics
```

debug firewall all

Enables / disables firewall debugging.

syntax:

[no] debug firewall all

example:

To enable all debug commands on the firewall, enter:

```
Router>  debug firewall all
```

applicable models:

All models.

debug disable-firewall

Bypasses all the firewall policy checks. By default this feature is disabled.

syntax:

[no] debug disable-firewall

example:

:To disable all policy checks on the firewall, enter:

```
Router> debug disable-firewall
```

To enable policy checks on the firewall (if disabled previously), enter:

```
Router> no debug disable-firewall
```

applicable models:

All models.

debug firewall dos-protect

Enables / disables dos-protect check debugging for firewall.

syntax:

[no] debug firewall dos-protect

example:

:To debug Denial of Service attack checks on the firewall, enter:

```
Router> debug firewall dos-protect
```

applicable models:

All models.

debug firewall ip-reassembly

Enables / disables ip-reassembly debugging for firewall.

syntax:

[no] debug firewall ip-reassembly

example:

:To debug IP reassembly on the firewall, enter:

```
Router>  debug firewall ip-reassembly
```

applicable models:

All models.

debug firewall packet

Enables / disables packet debugging for firewall.

syntax:

[no] debug firewall packet

example:

:To debug packet processing on the firewall, enter:

```
Router>  debug firewall packet
```

applicable models:

All models.

show debug all

Shows the status disable-firewall and all other debug commands.

syntax:

show debug all

example:

To show debug status of **disable-firewall** and all other debug commands, enter:

```
Router/configure>  show debug all
                Component Name                              Status
                ============================================
                Debug ARP                                 Disabled
                Debug DHCP RELAY                          Disabled
                Debug DHCP SERVER                         Disabled
                Debug FRAME RELAY                         Disabled
                Debug MHU                                 Disabled
                Debug NAT                                 Disabled
                Debug PPP                                 Disabled
                Debug SSH                                 Disabled
                Debug VRRP                                Disabled
                Debug BGP                                 Disabled
                Debug OSPF                                Disabled
                Debug RIP                                 Disabled
                Debug PIM                                 Disabled
                Debug IGMP                                Disabled
                Debug TUNNEL                              Disabled
                Debug FIREWALL                            Disabled
                Debug DISABLE-FIREWALL                    Disabled
Router/configure>
```

applicable models:

All models.

show debug firewall

Shows the status of all the firewall debug commands.

syntax:

show debug firewall

example:

To show debug details for the firewall, enter:

```
Router/configure>  show debug firewall

                Debug level                                  Status
                ============================================
                Debug Firewall All levels                  Disabled
                Debug Firewall Ip-Reassembly               Disabled
                Debug Firewall Algs                        Disabled
                Debug Firewall Dos-Protect                 Disabled
                Debug Firewall Packet                      Disabled
Router/configure>
```

applicable models:

All models.

show firewall connections

View the connection database.

syntax:

show firewall connections map-name [summary] [address] [port] [protocol]

parameter	definition
map-name	For a specific map - the name of that map. To name all maps, specify **all**.
summary	Only summary connection details.
address	The connections based on the specified IP address.
port	The connections based on the specified port numbers. Valid port numbers are 1—65535.
protocol	Shows connections based on the protocol specified, for example, UDP, TCP, and ICMP.

example:

To view all the connections for just the map **corp**, enter:

```
Router>  show firewall connections corp summary
TCP         UDP         ICMP        Other       Total       Max
---         ---         ----        -----       -----       ---
0           0           0           0           0           1024
```

To view all the connections in all the maps, enter:

```
Router>  show firewall connections all summary
Map Name         TCP         UDP         ICMP        Other       Total       Max
--------         ---         ---         ----        -----       -----       ---
self             0           0           0           0           0           216
internet         0           0           0           0           0           3072
corp             0           0           0           0           0           1024


Total Tcp Conns  0           Total Udp Conns   0
Total Icmp Conns 0           Total Other Conns 0
Router>
```

applicable models:

All models.

show firewall dos-protect

Displays the configured DoS-protect attack check settings.

syntax:

show firewall dos-protect

example:

:To see the DoS protection configuration, enter:

```
   Router>  show firewall dos-protect
DOS attack check                          Status
----------------                          ------
syn flooding attack check                 enabled
source route attack check                 enabled
winnuke attack check                      disabled
ftp bounce attack check                   disabled
icmp error attack check                   enabled
ip unaligned time stamp check             disabled
ip sequence number prediction check       disabled
ip sequence number range check            disabled
mime flood attack check                   disabled
Router>
```

applicable models:

All models.

show firewall interface

View the interfaces that are configured for security.

syntax:

show firewall interface <all | map-name>

parameter	definition
map-name	For a specific map - the name of that map. To name all maps, specify **all**.

example:

To view all the interfaces that are configured for security on the router, enter:

```
Router>  show firewall interface all

Interface      Map Name
---------      --------
ethernet1      corp
wan1           internet

Router>
```

To view all the interfaces that are configured for security in the map **corp**, enter:

```
Router>  show firewall interface corp

Interface      Map Name
---------      --------
ethernet1      corp

Router>
```

applicable models:

All models.

show firewall ip-reassembly

Displays the IP reassembly configuration.

syntax:

show firewall ip-reassembly

example:

To display the IP reassembly configuration, enter:

```
Router>  show firewall ip-reassembly
ip reassembly status         enabled
ip reassembly packet size    65535 bytes
fragment size in ip packet   28 bytes
ip reassembly fragment count 44
ip reassembly timeout value  60 seconds

Statistics
----------
ip reassembly requests              0
ip reassembly fails                 0
ip reassembly timeout fails         0
ip reassembly fragment count exceeds 0
ip reassembly packet size exceeds    0
Router>
```

applicable models:

All models.

show firewall logging

Displays configured logging thresholds.

syntax:

show firewall logging

example:

:To show current logging thresholds, enter:

```
Router>  show firewall logging
Logging info    Number of events
------------    ----------------
attack log      100
policy log      100
vpn log         100

Router>
```

applicable models:

All models.

show firewall max-connection-limit

Displays the maximum number of connections originating from a given map.

syntax:

show firewall max-connection-limit <all | int2self >

example:

To display the number of connections for all, enter:

```
Router>  show firewall max-connection-limit all
Map Name Connections
-------- -----------
self            216        0
Internet 3072
corp 1024
*int2Self 108
Total Configured:    4420
Total Supported:     29912

*int2self - As a special case, this indicates the max TCP connections
allowed from internet to self
```

applicable models:

All models.

show firewall nat-failover

Displays the configured NAT backup interfaces.
1show firewall nat-failover

syntax:
show firewall nat-failover

example:

To view the configured NAT backup interfaces for all the primary interfaces, enter:.
Router> **show firewall nat-failover**

applicable models:

All models.

show firewall nat-translations

Displays NAT (Network Address Translation) details.

syntax:

show firewall nat-translations map-name[address] [port][protocol]

parameter	definition
map-name	For a specific map - the name of that map. To name all maps, specify **all**.
address	Show the connections based on the specified IP address.
port	Show the connections that are using the specified port numbers. Valid port numbers are 1—65535.
protocol	Shows the connections based on the protocol specified, for example, UDP, TCP.

example:

To see NAT information for all maps, enter:

```
Router>  show firewall nat-translations all
```

To view the NAT translations for the map **corp**, enter:

```
Router>  show firewall nat-translations corp
```

applicable models:

All models.

show firewall object

Displays the objects that are configured on the router.

syntax:

show firewall object object-type map-name [object-name]

parameter	definition
object	The type of object. Objects can be: ftp-filter -- ftp-filter http-filter -- http-filter rpc-filter -- rpc-filter smtp-filter -- smtp-filter nat-pool -- nat-pool service -- service schedule -- schedule address -- address
map-name	For global - the name of the keyword for global objects. For map - the name of an object.
object-name	The name of the object.

example:

:To view all the configured ftp-filter objects for the map **corp**, enter:

```
Router>  show firewall object ftp-filter corp
Object Name     Action Log Commands
-----------     ------ --- --------
myftp           permit no  put get ls

Router>
```

example:

To view all the configured ftp-filter objects for global, enter:

```
Router>  show firewall object ftp-filter global
Object Name     Action Log Commands
-----------     ------ --- --------
gloftp            permit no  put
```

applicable models:

All models.

show firewall policy

Displays the firewall policy configuration.

syntax:

show firewall policy map-name [priority] [statistics] [detail]

parameter	definition
map-name	The name of the map.
priority	The firewall policy priority value. Valid range is 1—1024.
statistics	The statistical information available only for the firewall policy.
detail	All of the information available for the firewall policy.

example:

To view all the **corp** map firewall policies in brief mode, enter:

```
Router>  show firewall policy corp
Advanced: S - Self Traffic, F - Ftp-Filter, H - Http-Filter,
          R - Rpc-Filter, N - Nat-Ip/Nat-Pool, L - Logging,
          E - Policy Enabled, M - Smtp-Filter

Pri  Dir Source Addr        Destination Addr   Sport Dport Proto Action
Advanced
---  --- -----------        ----------------   ----------------- ------
--------
1022 out any               any               any   any   any   PERMIT SE
1023 in  any               any               any   any   any   PERMIT SE
1024 out any               any                any   any   any   PERMIT E
Router
```

To view all the **corp** map firewall policies in detail mode, enter:

```
Router>  show firewall policy corp detail
Policy with Priority 1022 is enabled, Direction is outbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Bytes In 0 Bytes Out 0

Policy with Priority 1023 is enabled, Direction is inbound
Action permit, Traffic is self
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Bytes In 0 Bytes Out 0

Policy with Priority 1024 is enabled, Direction is outbound
Action permit, Traffic is transit
Logging is disable
Source Address is any, Dest Address is any
Source Port is any, Dest Port is any, any
Schedule is disabled, Ftp-Filter is disabled
Smtp-Filter is disabled, Http-Filter is disabled
Rpc-Filter is disabled, Nat is disabled
Max-Connections 1024, Connection-Rate is disabled
Policing is disabled, Bandwidth is disabled
Bytes In 0 Bytes Out 0
Router>
```

applicable models:

All models.

show firewall port-trigger

Views the configured port-trigger records on the box.

syntax:
`show firewall port-trigger <port-trigger-name>`

parameter	definition
port- trigger- name	The name string of the port-trigger.

example:

To see specific the specific port-trigger record app1, enter:

```
Router> show firewall port-trigger app1
```

To see all the port-trigger records, enter:

```
Router> show firewall port-trigger
```

applicable models:

All models.

show firewall statistics

Displays the firewall statistics.

syntax:

statistics

example:

To display the firewall statistics, enter:

```
Router>  show firewall statistics
Src Addr Broadcast       0            Dest Addr Broadcast       0
Spoofed Packets          0            Yank Header Error         0
Access Deleted Policy    0            Data Without Connection   0
Inbound Policy Not Found 0            Outbound Policy Not Found 0
Invalid Src Interface    0            Invalid Dest Interface    0
Invalid Tcp Request      0            Invalid Udp Echo Reply    0
Invalid Icmp Error Msg   0            Invalid Icmp Echo Reply   0
Invalid Ack Value        0

Packets Originated from Self IP Stack
-------------------------------------
IP Packets               0            Non IP Packets            0

Incoming Packets
----------------
IP Packets/Frags         0            ARP Packets               0
Packets To Self          0            Packets Transmitted       0

Packets Frag And Send
---------------------
Fragments OK             0            In Discards               0
In Header Errors         0            Forward Datagrams         0
Out No Route             0            Out Discards              0
Fragments Fail           0            Fragments Created         0
Router>
```

applicable models:

All models.

show firewall stealth-mode

Displays the stealth-mode configuration.

syntax:

stealth-mode

example:

To display stealth-mode details, enter:

```
Router>  show firewall stealth-mode
Map Name        Stealth-mode
--------        ------------
internet        disable
corp            disable

Router>
```

applicable models:

All models.

show firewall timeout

View the timeout values for protocols and services.

syntax:

show firewall timeout [general | service-name]

example:

To display general timeout information, enter:

```
Router>  show firewall timeout general
Service Name    Timeout
------------    -------
tcp             600
udp             60
icmp            60
tcp-reset       20
ftp-inactivity  600
dns-inactivity  120
Router>
```

To display timeout information for a particular, non-general service (**timetcp**), enter:

```
Router> show firewall timeout timetcp
```

applicable models:

All models.

show firewall url-key-filter

Displays the keywords configured for blocking in the URLs.

syntax:

show firewall url-key-filter

example:

To display key names for URL filtering, enter:

```
Router>  show firewall url-key-filter
URL Key filtering enabled
URL Keys
--------
games
movies
Router>
```

21

VIRTUAL ROUTER REDUNDANCY PROTOCOL

The Router implementation of Virtual Router Redundancy Protocol (VRRP) conforms to the standard implementation of the protocol as specified in RFC 2338 and RFC 3768. As such, Router's use of VRRP is compatible with other vendors who implement VRRP per the specifications.

configure interface ethernet vrrp

This command configures a VRRP group for an Ethernet interface.

parameter	definition
group	Group number The range is 1 - 255.

syntax:

vrrp group < n >

example:

Router/configure/interface/ethernet 0> **vrrp 10**

next-level commands

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All systems.

configure interface ethernet vrrp advertisement_interval

This command configures the time interval for VRRP advertisements in seconds.

parameter	definition
adv_interval	Advertisement interval in seconds The range is 1 - 3600; the default is 1.

syntax:

advertisement_interval adv_interval < n >

example:

Router/configure/interface/ethernet 0/vrrp 10> **advertisement_interval 360**

related commands:

configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All systems.

configure interface ethernet vrrp authentication

This command configures the VRRP authentication information.

Once configured, all outgoing VRRP packets will have this authentication information and all packets received will be authenticated using this information.

parameter	definition
auth_string	Authentication string Enter a word (maximum of eight characters).

syntax:

authentication auth_string < auth_string >

example:

Router/configure/interface/ethernet 0/vrrp 10> **authentication alfxitz**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All systems.

configure interface ethernet vrrp description

This command assigns a description to the VRRP group.

parameter	definition
desc_string	Description string describing group Enter a string up to 80 characters within quotation marks.

syntax:

description < "desc_string" >

example:

Router/configure/interface/ethernet 0/vrrp 10> **description "virtual router for wan"**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All systems.

configure interface ethernet vrrp enable

This command enables a VRRP group.

syntax:

enable

example:

Router/configure/interface/ethernet 0/vrrp 10> **enable**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All systems.

configure interface ethernet vrrp ipaddr

This command configures VRRP group virtual IP addresses.

syntax:

ipaddr < ip address >

example:

Router/configure/interface/ethernet 0/vrrp 10> **ipaddr 128.44.10.24**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All systems.

configure interface ethernet vrrp learn_adv_internal

This command configures the backup router to learn the advertisement interval from the master.

syntax:

learn_adv_interval

example:

Router/configure/interface/ethernet 0/vrrp 10> **learn_adv_interval**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All systems.

configure interface ethernet vrrp preempt

This command configures the virtual router to prempt the current VRRP master if it has a higher priority than the current master.

syntax:

preempt

example:

Router/configure/interface/ethernet 0/vrrp 10> **preempt**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp priority
configure interface ethernet vrrp track

applicable models:

All systems.

configure interface ethernet vrrp priority

This command configures the priority level of the router within a VRRP group.

parameter	definition
level	Priority level The range is 1 - 255; the default is 100.

syntax:

priority level < n >

example:

Router/configure/interface/ethernet 0/vrrp 10> **priority 100**

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp track

applicable models:

All systems.

configure interface ethernet vrrp track

This command configures tracked interface and track priority.

parameter	definition
intfname	Interface name (e.g., ethernet0, ethernet1, or bundle name)
track_priority	Track priority The range is 1 - 255.

syntax:

track intfname track_priority < n >

example:

Router/configure/interface/ethernet 0/vrrp 10> **track ethernet1 140**

NOTE: An Ethernet interface cannot track itself. You must specify a different Ethernet interface to track. (See example above.)

related commands:

configure interface ethernet vrrp advertisement_interval
configure interface ethernet vrrp authentication
configure interface ethernet vrrp description
configure interface ethernet vrrp enable
configure interface ethernet vrrp ipaddr
configure interface ethernet vrrp learn_adv_internal
configure interface ethernet vrrp preempt
configure interface ethernet vrrp priority

applicable models:

All systems.

configure interface ethernet vrrp_mode

This command configures VRRP mode.

parameter	definition
mode	vrrp mode
0	VRRP mode 0 (Gratuitous ARP for an interface) This is the default.
1	VRRP mode 1 (Active/Standby)
2	VRRP mode 2 (Promiscuous mode)

syntax:

vrrp_mode mode < n >

example:

Router/configure/interface/ethernet 0/vrrp 10> **vrrp_mode 1**

applicable models:

All systems.

show vrrp

This command displays VRRP related information.

parameter	definition
group	VRRP group The range is 1 - 255; the default is all groups.
interface	
ethernet0	Ethernet 0 interface
ethernet1	Ethernet 1 interface
mode	
summary	Summary mode
detailed	Detailed mode

syntax:

vrrp [group < n >] [interface < interface >] [mode < mode >]

example:

Router> **show vrrp detailed**

screen display example

```
> show vrrp detailed
Ethernet 1
    VRRP Mode: 0 (Gratuitous Arp)
    VRRP Group:  10
        Description:
        State: MASTER Priority: 100  Preempt:  on Advertisement:    1 secs
        Authentication Type: No Authentication
        Virtual IP Addresses:
            IP Address 1: 128.44.10.24

        VRRP Router Up Time: 0 Days. : 0 Hours. : 0 mins. : 42 secs.
        Group Master IP Address: 192.168.70.222
    Statistics:
      Became Master: 1   Advertisements Received: 0
      Priority zero packets - Received : 0   Sent: 0
      Checksum errors: 0  Version Errors: 0   VRID errors:  0
      Advertisment Interval errors: 0  IP TTL  Errors: 0   Packet Type Errors: 0

      Auth Failures: 0  Invalid Auth Types: 0  Mismatched Auth Types: 0
      Address List mismatches: 0  Packet length Errors: 0

>
```

applicable models:

All systems.

22
ALARMS AND STATISTICS

This appendix provides definitions for the several types of alarms and statistical information that users need for performance monitoring and network conditions. Refer to the Commands chapter for information on configuring and displaying alarms and statistics.

Alarms

Type	Description
RAIS	Receive Alarm Indication Signal.
RLOF	Receive Loss of Frame.
LORC	Loss of Receive Clock.
ROOF	Receive Out of Frame. System is receiving M-frame and or M-subframe bit errors.
RRAI	Receive Remote Alarm Indication.
RFEBE	Receive Far End Block Errors.
TAIS	Transmit Alarm Indication Signal. This signal indicates a problem with the Router system.
RIS	Receive Idle Signal. System has detected an idle signal from the far end. This alarm indicates that the far end system is in a maintenance mode. The idle signal is defined in GR-499-Core, Section 10.5.1.3.
RPE	Receive P-bit Error. The P-bits are defined in GR-499-Core, Section 10.5.1.
TRAI	Transmit Remote Alarm Indication. System has detected a loss of incoming signal or Loss of Frame, causing it to send a Yellow Alarm signal to the far end. This Yellow Alarm code is the same as the RRAI code described above.
RLOS	Receive Loss of Sync. System has detected an OOF or Loss of Carrier condition. This alarm clears upon system resynchronization, which can only occur after the RLOC alarm clears.
TFBE	Transmit FEBE. System has sent an FEBE signal to the far end.

Type	Description
RAIS	Receive Alarm Indication Signal. System is receiving unframed all-ones from the far end.
RLOF	Receive Loss of Frame (system cannot frame up on incoming signal from the far end).
RRAI	Receive Remote Alarm Indication (far end system has detected a Loss of Signal or Loss of Frame, and sent a Yellow Alarm signal to the Router system). For an ESF-framed signal, this alarm is sent over the FDL as either an ANSI T1.403 or AT&T TR-54016 code. For a D4-framed signal, this alarm code is a zero bit as the second bit of every incoming DS0 channel (B2 = 0). This alarm clears when a logic 1 bit is detected in any DS0 channel.
TAIS	Transmit Alarm Indication Signal (system has sent an unframed all-ones signal to the far end instead of a normal T1 signal). This signal tells the far end that a problem exists in the path toward that end.
ROOF	Receive Out of Frame (system is receiving frame bit errors in 2 of 4, 2 of 5, 2 of 6, or 3 of 5 framing bits, per AT&T OOF state defined in TR-62411, Section 7.2). This alarm clears when the system reframes successfully on the incoming signal.
RLOS	Receive Loss of Sync (system has detected an OOF or Loss of Carrier condition). This alarm clears upon system resynchronization, which can only occur after the RLOC alarm clears.
LORC	Loss of Receive Carrier (system has received 175 consecutive zeros). This alarm clears upon reception of the next logic 1 bit (pulse).
TRAI	Transmit Remote Alarm Indication (system has detected a loss of incoming signal or Loss of Frame, causing it to send a Yellow Alarm signal to the far end). This Yellow Alarm code is the same as the RRAI code described above.

Type	Description
CA	DCE device is not available for data transmission.
LC	Remote Line Loopback C
ST	Send timing lost (No external timing clock was detected from the DCE.)
TA	DTE device is not available.
TM	Test mode request (DCE sent a request for the system to go into a test mode).
LALB	Loopback A; Loopback B

Statistics

Type	Description
CV	Coding Violations (excessive zeros or AMI bipolar violations). For a B8ZS-coded T1 signal, a CV is a bipolar violation that is not part of a valid zero-substitution code.
ES	Errored Seconds (number of seconds in which one or more CRC error events were detected). This count is not advanced during a UAS second.
SES	Severely Errored Seconds (number of 1-second intervals with 320 or more CRC error events, or one or more OOF events).
SAS	Severely Alarmed Seconds (number of seconds in which more than one AIS alarm was received).
UAS	Unavailable Seconds (number of 1-second intervals during which service is not available). The service is declared unavailable if 10 or more consecutive SESs occur.
LCV	Line Code Violations (REXZ states or AMI bipolar violations).
LES	A Line Errored Second is a second in which one or more CV occurred or one or LOS defects.
LSES	Line Severely Errored Seconds (number of 1-second intervals with 320 or more CRC error events, or one or more OOF events).

Type	Description
CV	Coding Violations (excessive zeros or AMI bipolar violations). For a B8ZS-coded T1 signal, a CV is a bipolar violation that is not part of a valid zero-substitution code.
ES	Errored Seconds (number of seconds in which one or more CRC error events were detected). This count is not advanced during a UAS second.
SES	Severely Errored Seconds (number of 1-second intervals with 320 or more CRC error events, or one or more OOF events).
SAS	Severely Alarmed Seconds (number of seconds in which more than one AIS alarm was received).
CSS	Controlled Slip Seconds (number of seconds in which the system replicated or deleted at least one T1 frame of data).
UAS	Unavailable Seconds (number of 1-second intervals during which service is not available). The service is declared unavailable if 10 or more consecutive SESs occur.
LCV	Line Code Violations (REXZ states or AMI bipolar violations).
LES	A Line Errored Second is a second in which one or more CV occurred or one or LOS defects.
LSES	Line Severely Errored Seconds (number of 1-second intervals with 320 or more CRC error events, or one or more OOF events).

Type	Description
EEV	ESF Error Events (running total number of ESF errors). An EEV is defined as the detection of a CRC error event described above, or the detection of an OOF event.
ES	Errored Seconds (number of seconds in which one or more EEVs were detected). This count is not advanced during a UAS second.
UAS	Unavailable Seconds (number of 1-second intervals during which T1 service is not available). The service is declared unavailable if 10 or more consecutive SESs are detected.
BES	Bursty Errored Seconds (number of 1-second intervals with more than 1 but less than 320 CRC error events).
SES	Severely Errored Seconds (number of 1-second intervals with 320 or more CRC error events, or one or more OOF events).
LOFC	Loss of Frame Count (number of times a LOF is detected). The alarm declaration and clearing times are defined in AT&T TR-54016.
CSS	Controlled Slip Seconds. Number of seconds in which the system replicated or deleted one or more T1 frames of data.

Type	Description
RLOS	Receive Loss of Signal status (ON=signal is lost).
RAIS	Receive alarm indication signal status (ON=AIS detected).
RLOF	Receive Loss of Frame status (ON= frame loss has occurred).
RRAI	Receive remote alarm status (ON= remote alarm received).
RIS	Receive idle signal status (ON=framed all-ones idle signal).
RFEBE	Receive far end block errors (ON=errors received).
RCPE	Receive C-bit parity errors (ON=parity errors received).
TAIS	Transmit alarm indication signal status (ON= AIS sent to far end).
TRAI	Transmit remote alarm status (ON= remote alarm sent to far end).
TFEBE	Transmit far end block errors (ON= errors sent to far end).
TLncod	Not currently used (always OFF).
TPlCod	Not currently used (always OFF).
TRstCod	Not currently used (always OFF).
TPtrn	Not currently used (always OFF).

Type	Description
PES	P-bit Errored Seconds. A PES is a second with one or more PCVs, one or more Out of Frame defects, or a detected incoming AIS. This gauge is not incremented when UASs are counted.
PSES	P-bit Severely Errored Seconds. A PSES is a second with 44 or more PCVs, or one or more Out of Frame defects, or a detected incoming AIS. This gauge is not incremented when UASs are counted.
SEFS	Severely Errored Framing Seconds. A SEFS is a second with one or more Out of Frame defects or a detected incoming AIS. This item is not incremented during unavailable seconds.
UAS	Unavailable Seconds (declared upon occurrence of 10 consecutive SESs).
LCV	Line Code Violations (number of excessive-zero occurrences detected in the incoming signal).

Type	Description
PES	P-bit Errored Seconds. A PES is a second with one or more PCVs, one or more Out of Frame defects, or a detected incoming AIS. This gauge is not incremented when UASs are counted.
PCV	P-bit Code Violations (P-bit parity errors).
LES	Line Errored Seconds (those with one or more coding violations or Loss of Signal).
CCV	C-bit Code Violations (C-bit parity errors)
CES	C-bit Errored Seconds (those with one or more coding CCVs, OOFs, or AIS detections).
CSES	C-bit Severely Errored Seconds (those with 44 or more CCVs, OOFs, or AIS detections).

Type	Description
ES	Errored Seconds. For ESF-framed T1 links, an ES is a second with one or more PCVs, OOFs, or AIS condition. For D4-framed links, an ES is a second with one or more bipolar violations. The ES count is not advanced during a UAS.
SES	Severely Errored Seconds.For ESF-framed links, a SES is a second with 320 or more PCVs or OOFs. Or, a SES is an AIS condition for ESF-framed T1s. For D4-framed T1 links, a SES is a second with framing error events, or 1544 or more LCVs on a D4-framed T1s). The SES count is not advanced during a UAS.
SEFS	Severely Errored Framing Seconds. A SEFS is a second with one or more Out of Frame defects or a detected incoming AIS. This item is not incremented during unavailable seconds.
UAS	Unavailable Seconds (declared upon occurrence of 10 consecutive SESs).
CSS	Controlled Slip Seconds. A CSS is a one-second interval containing one or more controlled slips. This is not incremented during an Unavailable Second.
PCV	P-bit Code Violations (P-bit parity errors).
LES	Line Errored Seconds (those with one or more coding violations or Loss of Signal).
BES	Bursty Errored Seconds (number of 1-second intervals with more than 1 but less than 320 PCVs, no SEF events, and no AIS signal detection). This count is not advanced during a UAS state.
DM	Degraded Minutes (those in which the incoming bit error rate is between 10^{-6} and 10^{-3}). These minutes are determined by collecting all available seconds, removing SESs, grouping the results in 60-second groups, and counting a 60-second group as degraded if the total errors during those seconds exceed 10^{-6}. Available seconds are those that are not UASs.
LCV	Line Code Violations (bipolar violations and excessive zeros.)

Type	Description
LCV	Line Code Violations (number of line code violations detected in the incoming signal).
FBE	Framing Block Errors (number of framing errors detected in the incoming signal).
PBE	P-bit Errors (number of P-bit parity errors detected in the incoming signal).
CPBE	CP-bit Errors (number of CP-bit parity errors detected in the incoming signal).
FEBE	Far end Block Errors. Number of times the received FEBE bits included a block error.
COFA	Change of Frame Alignment (number of times a frame realignment occurred in the incoming signal).

Type	Description
EEV	CRC Errors Events. Occur when the CRC-6 code calculated on the previous superframe does not match the code for the current superfame. These events are not counted during an RLOS alarm condition.
ES	Errored Seconds. Number of seconds in which one or more EEV's are detected. This count is not advanced during a UAS second.
UAS	Unavailable Seconds. Number of 1-second intervals during which T1 service is not available. The service is declared unavailable if 10 or more consecutive SES's are detected.
BES	Bursty Errored Seconds. Number of 1-second intervals with more than 1 but less that 320 CRC error events.
SES	Severely Errored Seconds. Number of 1-second intervals with 320 or more EEV's, or one or more OOF events.
LOFC	Loss of Frame Count. number of times a LOF is detected. The alarm declaration and clearing times are defined in AT&T TR-54016.
CSS	Controlled Slip Seconds. A CSS is a one-second interval containing one or more controlled slips. This is not incremented during an Unavailable Second.
BPV	Bipolar Violations (logic 1 pulses received with the same polarity as the preceding pulses from the T1 link.
OOF	Out of Frame occurrences. Number of times the system is in an OOF state.
CRC	Number of CRC-6 bit errors detected in the incoming signal.

T1 Module Commands

The following example shows how to display T1 statistics and alarms:

```
>  show module config t1 1

T1 1 is ENABLED
Alarm Hierarchy: TRUE,
Yellow Alarm: DISABLE
Framing:ESF, LineCode:B8ZS, ClockSource:LINE, LineMode:CSU, LBO:0
db
FDL: ANSI Unit Protocol enabled ,ATT Unit Protocol enabled ,
CsuDsuType:  CSU & DSU
CIRCUIT-ID : Not Configured ,CONTACT-INFO : Not Configured ,
DESCRIPTION : Not Configured ,

Line Status:
   RLOS:OFF    RAIS:OFF    RLOF:OFF    RRAI:OFF    TAIS:OFF
   TRAI:OFF  TLnCod:OFF  TPlCod:OFF TRstCod:OFF   TPtrn:OFF
   Loop:OFF    LORC:OFF
```

Additional commands are:

```
> show module userstats t1 1
> show module test t1 1
> show module alarms t1 1
```

T1 Bit Error Rate Test Commands

To isolate problems with a faulty T1 WAN link, perform line or payload loopbacks at either end of the link and perform a BERT test. This isolates a problem to either: the Router system, far-end equipment, interconnect cabling at either end or the T1 line between the two systems.

Loopback Test

To perform line and payload loopbacks at either end, use the appropriate command.

The following loopback commands are available:

```
test t1 1 loopback
```

BERT Test

Use this command to initiate a bit error rate test. The following BERT tests commands are available to test specific T1 links.

```
test t1 1 bert
```

View Bert Test

Once the test is started, view the results using the command:

```
/show/module/test>  t1 1
```

The following example shows how to gather BERT data:

First, raise the loopback on the T1 to test

`/test/t1 4` (This will use port 4 for example)

`/test/t1 4> loopback remote line` (raises the far end line loopback)

`/test/t1 4> bert interval 5` (runs BERT test for 5 minutes)

Now look at the bert test on T1 port 4:

```
/test/t1 4> show module test t1 4
```

You should see something like:)

```
Test Type:         BERT   Status:             LOCKED   Pattern:             QRW
Locked Seconds:       3   Pattern Loss Count:        0   Bit Error Count:     0

Configured Time:      2 minutes
Elapsed Time:         0 min.  3 sec.
```

After the test is completed, bring down the loopback:

`/test/t1 4> no loopback remote line` (drops the far end line loopback).

23
PPPoE

Use these commands to configure the PPPoE feature.

NOTE: PPPoE requires Release 8.3 or higher and is supported only on 600-series routers.

interface virtual-access

This command accesses the PPPoE virtual-access interface.

syntax:

interface virtual-access <name>

parameter	definition
name	The label given to this interface.

example:

Router/configure>**interface virtual-access test**

next-level commands

show interface virtual-access

applicable models:

OmniAccess 601, OmniAccess 602, and OmniAccess 604

ip negotiated

This command specifies that the IP address for the interface is obtained via PPP/IPCP (IP Control Protocol) address negotiation with the PPPoE server.

syntax:

ip negotiated

example:

Router/configure/interface/virtual-access <name>> **ip negotiated**

next-level commands

show interface virtual-access

ppp authentication

Authentication for PPP connection can be specified as PAP, CHAP, either, or none.

syntax:

ppp authentication <pap | chap | either | none> sent-username <username> password <password>

parameter	definition
PAP, CHAP	The type of authentication.
sent-username	The name of a user to be authenticated. Required for all authentication types except for `none`.
password	The password for this user. Required for all authentication types except for `none`.

example:

Router/configure/interface/ virtual-access <name>>ppp authentication chap

next-level commands

show interface virtual-access

applicable models:

OmniAccess 601, OmniAccess 602, and OmniAccess 604

pppoe ac-name

This command allows the user to configure a specific PPPoE server.

syntax:

interface virtual-access <name>

parameter	definition
name	The name of the access concentrator or PPPoE server used in PPPoE Active Discovery Offers (PADO).

example:

Router/configure/interface/ virtual-access <name>> **pppoe ac-name router1**

next-level commands

show interface virtual-access

applicable models:

OmniAccess 601, OmniAccess 602, and OmniAccess 604

pppoe ethernet

This command allows the user to specify the Ethernet interface on which PPPoE is enabled.

syntax:

pppoe ethernet<number>

parameter	definition
number	The number of the Ethernet interface.

example:

Router/configure/interface/ virtual-access <name>> **pppoe ethernet 1**

next-level commands

show interface virtual-access

applicable models:

OmniAccess 601, OmniAccess 602, and OmniAccess 604

protocol pppoe

This command specifies the tunneling protocol, with an optional parameter client.

syntax:

protocol pppoe <client>

parameter	definition
client	Specifies the client (optional).

example:

Router/configure/interface/ virtual-access <name>> protocol pppoe

next-level commands

show interface virtual-access

applicable models:

OmniAccess 601, OmniAccess 602, and OmniAccess 604

ppp keepalive

This command specifies the amount of time PPP should be keep up when there is no traffic.

syntax:

ppp keepalive <interval>

parameter	definition
interval	The time to keep PPP up.

example:

Router/configure/interface/ virtual-access <name>> ppp keepalive

applicable models:

OmniAccess 601, OmniAccess 602, and OmniAccess 604

24 ISDN

This chapter describes the commands used to configure the ISDN interface on 600-series routers.

NOTE: ISDN requires Release 8.3 or higher.

switch-type

Configures the ISDN service provider switch type. Use the **no** form of this command to configure the switch type to `basic-ni` (National ISDN switch type).

syntax:

Switch-type <switch-type>

parameter	definition
switch-type	Specifies the service provider switch type. Valid values are: basic-ni -- National ISDN Switch Type (default) basic-dms -- NT DMS-100 switch type basic-5ess -- AT & T basic rate switch type basic-1tr6 -- German 1tr6 switch type ntt -- ntt switch type vn3 -- French vn3 switch type basic-etsi -- ETSI basic switch type basic-ccitt -- CCITT basic switch type

example:

To configure the bundle "wan1" with ISDN switch type basic-5ess (National AT & T), execute the following command:

`Router/configure/interface bundle wan1/isdn/` **`switch-type basic-5ess`**

Switch type details are:

Table 8:

		User Side		Network Side	
Variant	**Country**	**BRI**	**PRI**	**BRI**	**PRI**
CCITT	World	Y	Y	Y	Y
4ESS	ATT - North America	N/A	Y	N/A	Y
5ESS	ATT - North America	Y	Y	Y	Y
NT DMS 100	Nortel - North America	Y	Y	Y	Y
VN2	France	Y	Y	Y	Y
VN3	France	Y	Y	Y	Y
NTT	Japan	Y	Y	Y	Y

Table 8:

ETSI	Europe	Y	Y	Y	Y
TR 303 - TMC	Bellcore - North America	N/A	Y	N/A	Y
TR 303 - CSC	Bellcore - North America	N/A	Y	N/A	Y
NT DMS 250	Nortel - North America	N/A	Y	N/A	Y
NT National ISDN	Nortel - North America	N/A	N/A	N/A	Y
NT DMS 250 Sprint Supplementary Services	Nortel - North America	N/A	Y	N/A	Y
NT MCI	Nortel - North America	N/A	Y	N/A	N/A
TR6 - PBX (PRI)	Germany	N/A	Y	N	N/A
TR6 - MPC (BRI)	Germany	Y	N/A	N	N/A
Australian Telecom	Australia	Y	Y	N	Y
NI-2	North America	N/A	Y	N/A	Y
NI-1	North America	Y	N/A	N	N/A
Q-SIG	World	N/A	Y	N/A	N/A
Q.932	ITU				
Keypad Protocol	CCITT	Y	Y	Y	Y
Feature Key Management	CCITT	Y	N/A	Y	N/A
Functional Protocol	CCITT	Y	Y	Y	Y

applicable models:

OmniAccess 601

bundle

Creates a logical bundle with "**bundle-name**". "**no**" form of this command will remove the logical bundle.

syntax:

bundle <bundle-name> <cr>

parameter	definition
bundle-name	Configures the bundle name, max 8 characters.

example:

To create a bundle "wan1", execute the following command:

```
Router/configure/interface bundle wan1
```

applicable models:

OmniAccess 601

link bri

Configures the bundle with BRI links with bandwidth option of 64 kbps or 128 kbps. This command is similar to T1 or E1 commands, which will configure the bundle with links. "**no**" form of this command will remove the specified links from the bundle.

syntax:

link bri <64/128> <cr>

parameter	definition
link-spec	BRI bandwidth - 64 or 128(enter a word).

example:

To add 64 kbps bri link to the bundle "wan1", execute the following command:

```
Router/configure/interface bundle wan1/link bri 64
```

applicable models:

OmniAccess 601

isdn

Configures the bundle with ISDN parameters. This is a command sub-tree.

syntax:

isdn

parameter	definition
isdn	configure isdn parameters

example:

To configure the ISDN related parameters for bundle "wan1", following command have to be executed to enter the ISDN sub-tree. Under ISDN sub-tree, related parameters can be configured

```
Router/configure/interface bundle wan1/isdn
```

applicable models:

OmniAccess 601

spid1

Configures the ISDN service profile identifier for BRI channel 1. SPID is used for defining the services subscribed by ISDN device. All the switch types don't support SPID. SPID is configured for each B Channel. SPID is usually a telephone number with some optional numbers. However, numbering schemes depends on service profile. The Local directory Number (LDN) is required only when two SPIDs are configured (for example, when connecting to a DMS-100 or NI1 switch). Configuring LDN is optional. **"no"** form of this command will unconfigure the SPID for BRI Channel 1.

syntax:

spid1 <spid-number> [ldn] <cr>

parameter	definition
spid-number	Specifies the service profile identifier (a word).
ldn	Specifies the local directory number. Valid range is 1-9999999.

example:

To configure the bundle "wan1" with SPID "1234567" with local directory number 4567, execute the following command:

```
Router/configure/interface bundle wan1/isdn/ spid1 1234567 ldn 4567
```

applicable models:

OmniAccess 601

spid2

Configures the ISDN service profile identifier for BRI channel 2. SPID is used for defining the services subscribed by ISDN device. All the switch types don't support SPID. SPID is configured for each B Channel. SPID is usually a telephone number with some optional numbers. However, numbering schemes depends on service profile. "**no**" form of this command will unconfigure the SPID for BRI channel 2.

syntax:

spid2 <spid-number> [ldn] <cr>

parameter	definition
spid-number	Specifies the service profile identifier (a word).
ldn	Specifies the local directory number. Valid range is 1-9999999.

example:

To configure the bundle "wan1" with SPID "7654321" with local directory number 7654, execute the following command:

```
Router/configure/interface bundle wan1/isdn/ spid2 7654321 ldn 7654
```

applicable models:

OmniAccess 601

keep-alive

Configures the Q.921 keep alive time. Use the **no** form of this command to set the Q.921 keep alive time to 10000 ms.

syntax:

keep-alive <time> <cr>

parameter	definition
time	The keep-alive time in milli-seconds. The default is 10000 ms. Valid range is 6000 - 60000 ms.

example:

To configure the LAPD keep alive time to 6000 milliseconds, execute the following command:

```
Router/configure/interface bundle wan1/isdn/keep-alive 6000
```

applicable models:

OmniAccess 601

tei

Determine when TEI negotiation occurs. TEI negotiation can occur on power up or on first call. TEI negotiation is useful for switches that may deactivate Layers 1 or 2 when there are no active calls. **"no"** form of this command will set TEI negotiation on **powerup**.

syntax:

tei tei-type <cr>

parameter	**definition**
tei type	Specifies the tei type.

example:

To configure the bundle "wan1" with tei negotiation on first call, execute the following command:

```
Router/configure/interface bundle wan1/isdn/ tei first-call
```

NOTE: Only tei negotiation on powerup is supported in Release 8.3.

:

OmniAccess 601

caller

Configures the expected origin of call. All calls other than these origins or calls without caller-id are rejected. Caller ID Check adds a level of security for incoming calls. This command can be called multiple times.

Note that, when ISDN device is connected to ISDN switch, which doesn't supply caller ID, all calls will be rejected.

"**no**" form of this command will make all calls acceptable without Caller ID Check.

syntax:

caller <caller-number> <cr>

parameter	definition
caller-number	The caller number. The number is limited to a maxiumum of 20 digits.

example:

To configure the bundle "wan1" with caller ID "1234567", execute the following command:

```
Router/configure/interface bundle wan1/isdn/ caller 1234567
```

NOTE: Multiple caller configurations are planned for a future release.

:

OmniAccess 601

callednum

Configures the number to be dialed. This number will be used to call, whenever interesting data will flow through the BRI bundle. "**no**" form of this command will remove the configured called number.

syntax:

callednum <called-numbe> [sub-address] <cr>

parameter	definition
callednum	The number to be dialed. The number (entered as a word) is limited to a maximum of 10 digits.
sub-address	A number limited to a maximum of 10 digits.

example:

To configure the bundle "wan1" with called number "1234567", execute the following command:

```
Router/configure/interface bundle wan1/isdn/ callednum 1234567
```

applicable models:

OmniAccess 601

answer1

Configures the called party number or sub address in the incoming setup message. Only single device will answer the call by verifying the called party number or sub address or both. Maximum of 10 alphanumeric characters are permitted for called party number. "**no**" form of this command will disable this feature.

syntax:

answer1 [called-party-number] [sub-address]<cr>

parameter	definition
called-party-number	The number to be dialed. The number (entered as a word) is limited to a maximum of 10 digits.
sub-address	A number limited to a maximum of 10 digits.

example:

To configure the bundle "wan1" with called party number "1234567", execute the following command:

```
Router/Configure/interface bundle wan1/isdn/ answer1
called-party-number 1234567
```

To configure the bundle "wan1" with called party number "1234567" & with sub-address 7890, execute the following command:

```
Router/Configure/interface bundle wan1/isdn/ answer1
called-party-number 1234567 sub-address 7890
```

To configure the bundle "wan1" with sub-address 7890, execute the following command:

```
Router/Configure/interface bundle wan1/isdn/ answer1 sub-address 7890
```

To configure the bundle "wan1" with range of called party numbers from "1234560" to "1234569", execute the following command:

```
Router/Configure/interface bundle wan1/isdn/ answer1
called-party-number 123456X
```

Letter X is the wildcard, whenever specified, will be replaced by the range of numbers 0 to 9.

NOTE: Wild card support is planned for a future release.

applicable models:.

OmniAccess 601

answer2

Configures the second called party number or sub address in the incoming setup message. Only single device will answer the call by verifying the called party number or sub address or both.

Maximum of 10 alphanumeric characters are permitted for called party number. "**no**" form of this command will disable this feature.

syntax:

Answer2 [called-party-number] [sub-address]<cr>

parameter	**definition**
called-party-number	The number to be dialed. The number (entered as a word) is limited to a maximum of 10 digits.
sub-address	A number limited to a maximum of 10 digits.

example:

To configure the bundle "wan1" with called party number "1234567", execute the following command:

```
Router/configure/interface bundle wan1/isdn/ answer2
called-party-number 1234567
```

To configure the bundle "wan1" with called party number "1234567" & with sub-address 7890, execute the following command:

```
Router/configure/interface bundle wan1/isdn/ answer2
called-party-number 1234567 sub-address 7890
```

To configure the bundle "wan1" with sub-address 7890, execute the following command:

```
Router/configure/interface bundle wan1/isdn/ answer2 sub-address 7890
```

To configure the bundle "wan1" with range of called party numbers from "1234560" to "1234569", execute the following command:

```
Router/configure/interface bundle wan1/isdn/ answer1
called-party-number 123456X
```

Letter X is the wildcard, whenever specified, will be replaced by the range of numbers 0 to 9.

applicable models:

OmniAccess 601

disconnect-cause

Configures the disconnect cause code. Configured cause code will be sent to the switch, when application fails to complete the call. "**no**" form of this command will set the disconnect-cause to busy.

syntax:

disconnect-cause cause-code <cr>

parameter	definition
cause-code	The code in string.

example:

To configure the bundle "wan1" with disconnect-cause to not available, execute the following command:

```
Router/configure/interface bundle wan1/isdn/disconnect-cause
not-available
```

applicable models:

OmniAccess 601

chap username

Configures the CHAP user name and password. CHAP is Challenge Handshake Authentication Protocol, used during PPP negotiations. Authentication phase will succeed only when valid username and password are configured. "**no**" form of this command will disable the CHAP configurations.

syntax:

<username> password <password> <cr>

parameter	**definition**
COMMANDS	Any of the following commands can be used username -- configures chap username for PPP/MLPPP bundle password -- configures chap password for PPP/MLPPP bundle

example:

To configure the bundle "wan1" with CHAP username "alcatel" password "switch" :

```
Router/configure/interface bundle wan1/ chap username an password
switch.
```

NOTE: Password will be visible in clear text, when command is typed. But password will be displayed based on secure password feature configuration during "show configuration running".

applicable models:

OmniAccess 601

idle-timeout

Configures the idle timeout Value. ISDN connection will get automatically disconnected after idle-timeout period, if there is no traffic passing for the specified duration. This will be very handy command since ISDN connectivity is charged on the basis of usage. Use the **no** form of this command to configure the idle timeout to 5 minutes. Configure the timeout to 0 to disable this feature and set the ISDN connection to act as a leased line.

syntax:

idle-timeout <timeout> <cr>

parameter	definition
timeout	The timeout to disconnect the connection (default: 5 minutes). Range is 0– 60 minutes. Specifying 0 disables this feature.

example:

To configure ISDN idle timeout for 10 minutes, execute the following command:

```
Router/configure/interface bundle wan1/ isdn/ idle-timeout 10
```

NOTE: Due to the performance tuning, the idle-timer may not disconnect exactly at the specified timeout value.

applicable models:

OmniAccess 601

connect-delay

Configures the connect-delay for ISDN bundles. This is a very handy command when primary interface flaps. The configured timeout will make sure that ISDN bundle is not connected for specified seconds, from the first packet arrived on the ISDN interface. If the data continues to flow over ISDN interface for connect-delay period, then the call will be triggered.

For example, when primary interface goes down, first packet landing over ISDN interface will trigger a internal timer for "connect-delay". If the data continues to pass over ISDN interface for connect-delay, then the ISDN call is triggered to remote end.

After the first packet, if there is no data for (connect-delay+30secs), then call will triggered only if the data transmission over the ISDN interface continues for another "connect-delay" period.

"**no**" form of this command will configure the connect-delay to 15 seconds

syntax:

connect-delay **<delay>** <cr>

parameter	**definition**
timeout	Specifies the connect delay in seconds (default: 15 secs). Valid Range(s) : 1 - 60.

example:

To configure ISDN connect-delay for 10 seconds, execute the following command:

```
Router/configure/interface bundle wan1/ isdn/ connect-delay 10
```

NOTE: ISDN bundles can be brought down administratively using "Shutdown" command similar to any other bundles. If call is active, "shutdown" will tear down the call.

applicable models:

OmniAccess 601

show interface bundle

Existing show interface bundle command will display the ISDN bundle related information.

In this command, along with the Encapsulation, IP Address, counter details, ISDN related information such as TEI, Q921 keep-alive, Calling numbers, caller number, disconnect cause code, numbers to answer will be displayed.

syntax:

show interface bundle < bundle-name >

applicable models:

OmniAccess 601

show interface bundles

Existing show interface bundles command will display the ISDN bundle related information.

applicable models:

OmniAccess 601

show isdn global

Displays the switch-type configured and ISDN BRI interfaces on the router.

syntax:

show isdn global <cr>

parameter	definition
global	Displays ISDN global information.

example:

To display ISDN global information, enter:

```
Router> show isdn global
```

applicable models:

OmniAccess 601

show isdn interfaces

Displays BRI interface information such as TEI, Q921 keep-alive, Calling numbers, caller number, disconnect cause code, numbers to answer will be displayed for all BRI interfaces.

syntax:

interfaces <cr>

example:

To display all the BRI interfaces information, execute the following command:

```
router>show isdn interfaces
```

applicable models:

OmniAccess 601

show isdn interface

Displays the BRI interface information.

syntax:

show isdn interface <bundle-name> <cr>

parameter	definition
interface-name	The name of the bundle.
unreachable	Controls ICMP traffic processing for unreachable destinations.

example:

To display the BRI interface related information for bundle "isdn1", execute the following command:

```
router>show isdn interface isdn1
```

applicable models:

OmniAccess 601

show isdn statistics

Displays the BRI statistics information such as statistics details – tx, rx, bytes, errors for D channel & Bearer channels B1 & B2.

syntax:

statistics

example:

To display the BRI related, use the following command:

```
router>show isdn statistics
```

applicable models:

OmniAccess 601

show debug all

Existing show debug all command will display the ISDN debug status information along with the existing display.

applicable models:

OmniAccess 601

clear isdn statistics

Clears the BRI statistics information

syntax:

statistics <cr>

example:

To clear the BRI statistics, execute the following command:

```
router>clear isdn statistics
```

applicable models:

OmniAccess 601

debug isdn

Debugs ISDN messages, packets.

syntax:

COMMANDS <cr>

parameter	definition
COMMANDS	Includes any of the following: all—enables all debug messages for isdn messages cc—enables debug for call control messages q.921—enables q.921 debug messages q.921-timers—displays q.921 timers q.931— enables q.931 debug messages q.931-timers—displays q.931 timers physical-layer—enables physical layer messages on the console data-path—enables ISDN data path debug messages

example:

To enable all debug messages for ISDN, execute the following command:

```
router>debug isdn all
```

applicable models:

OmniAccess 601

debug ppp auth

Debug authentication phase in (ML) PPP.

syntax:

auth [interface-name] <cr>

parameter	definition
interface-name	The name of the bundle.

applicable models:

OmniAccess 601

GLOSSARY

Numerics

10 Base-T A specification describing a 10 Mbps Ethernet connection, typically set up via a twisted pair, Category 5 cable.

100 Base-T A specification describing a fast, 100 Mbps Ethernet connection, typically set up via a twisted-pair, Category 5 cable.

A

ACK Acknowledgment. A character transmitted by the receiver of data to acknowledge a signal or packet received from the sender.

AF Address Field. A sequence of bits that follow a packet's opening flag; these bits identify a secondary station that is either receiving or forwarding the packet.

Aggregation The bonding of one or more network connections to create a single higher-speed connection. Virtual multi-megabit access path is an example of aggregation technology. See also **VMAP**.

AIS Alarm Indication Signal. A signal sent downstream to indicate that a loss of signal has occurred upstream.

AMI Alternate Mark Inversion. A form of line coding in which zeros are represented by 01 (no amplitude) and ones are alternately represented by 00 and 11 (uniform amplitude).

ANSI American National Standards Institute. A voluntary organization, comprised of businesses, government, and other agencies, that develops standards relating to networking and communications. Provides a standard for the Facility Data Link.

ARP Address Resolution Protocol. A protocol used to translate an IP address to a lower level network, or host, address.

ASCII American Standard Code for Information Exchange. The code used by most computing devices to translate 128 letters, numbers, and symbols into 8-bit binary codes.

ATT An option for configuring the Facility Data Link. Under this option the errors are reported for the last 24 hours grouped by 15 minute intervals. Established by American Telephone and Telegraph (ATT&T).

AWG American Wire Gauge. Measures the diameter of non-ferrous conductors such as copper and aluminum. The higher the AWG rating is, the thinner the wire will be.

B

B8ZS Bipolar 8-Zero Substitution. A form of line-coding in which eight consecutive zeros are replaced by a special code that meets minimum ones-density requirements.

Bandwidth A term describing the size of a communications channel. Narrow (below 1.544 Mbps), wide (1.544 to 45 Mbps), and broad (above 45 Mbps) are examples of different bandwidths.

Bc Committed Burst Size. The maximum amount of data (in bits) that Frame Relay networks can send and receive. This amount is negotiable. See also **CIR**.

Be Excess Burst Size. The amount of data (in bits) that a Frame Relay network will attempt to send once Bc has been exceeded. This amount is negotiable. See also **Bc**.

BEC Bit Error Count. The number of unsuccessfully transferred bits. Compared against total bits transmitted to calculate the Bit Error Rate. See also **BERT**.

BECN Backward Explicit Congestion Notification Bit. A bit sent backwards along congested Frame Relay paths notifying senders of an information overload.

BER Bit Error Rate. The ratio of errored bits to successfully transmitted bits. BER is often expressed in powers of ten; ten to the sixth would represent one error for every million bits sent.

BERT Bit Error Rate Test. A test in which a known quantity and pattern of bits are transmitted, and errored bits are received and counted to calculate a BER.

BES Bursty Errored Seconds. Seconds containing errors (more than one code violation per second) that were received at the maximum data transmission rate.

BGP4 Border Gateway Protocol version 4. The current exterior routing protocol used to provide peer routers with addresses and next-hop information.

Bit Error A received bit with an incorrect logic state.

BNC Bayonet-locking Connector. A connector commonly found on slim Ethernet coaxial cables.

BOOTP Boot Protocol. A start-up protocol that enables network device to determine either its own or its ethernet interfaces' IP addresses.

BPS Bits Per Second. The rate at which data is transferred. In telecommunications BPS refers to bits per second, but in the computing industry often refers to bytes per second.

BPV Bipolar Violation. The presence of two consecutive “one” bits of the same polarity on a T1 line.

Bundle The grouping of multiple T1s in a Router system to create virtual multi-megabit access paths. In multiplexing, bundling refers to the creation of multiple groupings within a T1. See also **Aggregation**.

Burst Size Refers to the size of a single burst of data, in cells or packets.

Bursty Second A second during which at least two, but not more than 320, code violations occurred, and no out-of-frame states existed.

C

C/R Command Response Bit. A Frame Relay term describing a one-bit addition to a frame Address Field that enables communications with polled protocols such as SNA. See also **AF** and **SNA**.

Channel A path of communication, either electrical or electromagnetic, between two or more points. Can also be referred to as a circuit, line, link, or path.

CIR Committed Information Rate. A transmission rate that a Frame Relay network agrees not to exceed. Usage above the CIR could result in discarded information or increased fees.

CLI Command Line Interface. The line on the screen where commands are entered to configure and maintain a Router system.

Clock A signal that provides synchronized transmission with other devices and regulates the operations of a processor. Clocks use a quartz crystal to generate a uniform electrical frequency from which digital pulses are created.

Connection- Oriented A term used to describe network communication in which there are three well-defined phases: connection establishment, data transfer, and connection release. An example of a connection oriented protocol is Frame Relay.

Connectionless A term used to describe sending and receiving data without first establishing a connection, and without receiving immediate acknowledgment of receipt. TCP/IP is an example of a connectionless protocol.

CPE Customer Premises Equipment. All devices at a customer location that are connected to a communications network. Includes phones, PBXs, PCs, routers, etc.

CRC Cyclic Redundancy Check. A method of confirming error-free data. The transmitting device makes a calculation based on the frame being sent and appends this number to the frame; the receiving device double-checks the sender's calculation.

CSS Controlled Slip Seconds. Seconds in which a receiving terminal replicates or deletes groups of bits in order to compensate for a timing difference between the sending and receiving nodes.

CSU Channel Service Unit. An end-user box that terminates a T1 line at a customer's premises, and performs various coding, conditioning and equalization functions.

CV Code Violation. A violation of the format code used on the T1 line. In superframe mode, a CV represents any 3 ms interval with at least one BPV. In extended superframe mode, a CV represents any 3 ms time interval with at least one CRC error.

D

D4 A type of T1 signal framing; also called superframe (SF). One superframe consists of 12 frames, each containing a framing bit. The odd numbered framing bits are used for terminal synchronization, the even numbered bits are used for superframe alignment.

DCE Data Circuit-Terminating Equipment or Data Communications Equipment. Equipment located on the network end of a user-to-network interface that resolves interface problems between a Data Terminal Equipment and the network. See also **DTE**.

DE Discard Eligibility Bit. A bit attached to data that is in excess of the Committed Information Rate and eligible for discard. See also **CIR**.

DLCI Data Link Connection Identifier. A 10-bit portion of a frame relay header that identifies Data Link and PVC parameters such as frame size, Committed Information Rate, Bc, Be, and Tc. See also **CIR**, **Bc**, **Be** and **Tc**.

DNS Domain Naming System. The system by which numerical IP addresses are substituted for easily recalled Domain names such as Routernet.com.

DS0 Digital Signal, Level Zero. A 64 kbps increment of data on a DS1 signal. There are 24 DS0s in one DS1.

DS1 Digital Signal, Level One. A 1.544 Mbps digital signal carried on a T1 twisted-pair transmission line. Contains 24 DS0s.

DS3 Digital Signal, Level Three. A 43.7 Mbps digital signal carried via coaxial cables or fiber.

DSU Data Service Unit. A digital transmission device that performs in much the same way as a modem would for analog transmissions; it takes data from terminals, encodes it, sends it downstream and decodes it at the receiving end.

DSX Digital Signal Cross-Connect. A patch bay to which T1 and DS1 lines are wired that allows for cross-connections. See also **T1** and **DS1**.

DTE Data Terminal Equipment. End user equipment that sends and/or receives information. DTEs send their information to DCEs (Data Communication Equipment) for transfer to a network.

E

E1A-530 A high-speed serial interface specification.

EA Address Field Extension. Used in Frame Relay; a two bit portion of the address field that signifies that the address field has been extended beyond the standard two octets.

EER Excessive Error Rate. A framed T1 signal with an error rate exceeding the user's set threshold is being received.

EEV Excessive Error Violations. Error count has exceeded the allowable limit and has caused an excessive error violation.

Encapsulation The technique by which each subsequent layer in an internet protocol stack appends it header before transmission to the next layer. Encapsulation can also refer to frames from one protocol being stored as data in another.

ES Errored Second. A second in which at least one code violation was detected on a digital circuit.

ESD Electrostatic Discharge. The discharge of a static charge on a body or surface through a conductive path (equipment) to the ground. Potentially damaging to integrated circuits.

ESF Extended Superframe. A T1 signal consisting of two superframes (24 framing bits). Half the framing bits comprise a facilities data link (FDL) for maintenance and T1 performance monitoring. See also **FDL**.

Ethernet A Local Area Network used to connect computers and peripherals that are located within a small geographic area. Ethernet LANs can commonly transmit and receive data at rates of 10 or 100 Mbps. See also **LAN**.

F

FCS Frame Check Sequence. A 16-bit field that contains transmission error checking information. The FCS is usually found at the end of a frame

FDL Facilities Data Link. A 4 Kbps overhead channel used for in-service monitoring and diagnostics. The FDL consists of every other framing bit of an extended superframe.

FEAC Far End Alarm Control.

FECN Forward Explicit Congestion Notification. Term associated with Frame Relay. The FECN bit is located in the Address field and notifies a receiving device that the transmission has encountered congestion on the network.

Flash Memory The component in the Router system that stores configuration data and software programs. Flash memory can be user-updated for system upgrades and maintenance purposes.

Fractional T1 The provisioning and use of a T1 line's 24 channel subset.

Frame Relay Industry-standard switched data link layer protocol that can transport packets of varied length over multiple virtual circuits using HDLC encapsulation.

Framing A procedure in which overhead bits are inserted into a digital signal so that the receiver can recognize time slots corresponding to the separation of multiplexed channels on a T1.

FTP File Transfer Protocol. An application protocol used in the TCP/IP networking architecture to transfer files between network nodes.

Full-Duplex Simultaneous transmission of data upstream and downstream on one line.

G

GMT Greenwich Mean Time. The internationally recognized time for internet communications. GMT is based on the local time zone of the Greenwich Observatory in London, England.

GND Ground. An equipment connection to a wire that ends up in the earth. Ground connections draw excess voltages away from sensitive circuits that could otherwise be damaged.

H

H.323 Standards set up by the International Telecommunications Union that govern packet-based networking such as IP. H.323 sets the standard for real-time transmission of voice and video data.

Half-Duplex A circuit which can transmit data upstream and downstream, but cannot do both at the same time.

HDLC High-Level Data Link Control. A data link layer protocol that places control information in a specific place and specifies different bit patterns for control data; this encapsulation method reduces the number of errored bits.

Hello Message transmitted to the next node to set up a Multi-link Frame Relay connection. The retry or retransmission interval is set by the hello timer.

HSSI High-Speed Serial Interface. A network standard used for serial connections of up to 52 Mbps for WAN access.

HTTP HyperText Transfer Protocol. A protocol that allows a Web server and client browser to communicate and move documents around the Internet; HTTP is transparent to the web user.

I

IC Interface Card. A module in the Router system that contains its WAN, HSSI, and other interface circuits.

ICMP Internet Control Message Protocol. A diagnostic tool that can perform many test functions and report problems to hosts. It is an integral part of IP.

IEEE Institute of Electrical and Electronics Engineers. A technical society that also functions as a standards making body, responsible for many telecom and computing standards.

IETF Internet Engineering Task Force. A volunteer group that establishes and maintains new Internet TCP/IP standards.

IGMP Internet Group Management Protocol. Used by IP hosts to report their multicast group memberships to an adjacent multicast router.

IND Indication. Lights, beeps, or buzzers that indicating that something has or is about to happen. A green power LED indicates the Router system is receiving power.

Internet The world's largest and most comprehensive computer network. The Internet is a network of networks connecting tens of thousands of networks worldwide with T1 and other backbone links.

IP Internet Protocol. The network layer protocol of the TCP/IP network architecture that provides for connectionless transfer of packets. IP recognizes incoming messages, keeps track of node addresses, and allows a packet to traverse multiple networks.

IP Broadcast Internet Protocol Broadcast. Sending packets to all hosts on a single IP address. Sent on the same “best effort” basis as a regular IP unicast with no guarantees as to the order, time, or reliability of the transfer.

IP Mask Internet Protocol Mask. The 32-bit mask that is used in IP to identify the portion of the IP address that is representing the subnet address. See also **Subnet Mask**.

IP Multicast Internet Protocol Multicast. Sending packets to one or more hosts on a single IP address. Like standard IP packet switching, there is no guarantee on reliability, sequence, or arrival time.

IP MUX IP Multiplexing. Splitting a channel into two or more divisions, each capable of running the IP network layer protocol.

IPC Interprocess Communications. Used to offer services to and receive services from other programs. IPC allows threads and processes to transfer messages among themselves.

ISP Internet Service Provider. A company that provides access to the Internet for corporate and private customers. ISPs can be reached either by dial-up connections or dedicated lines, and usually provide customers with a suite of services.

ITC Inter-Task Communications. The capability of two programs to share information. Allows programs to update themselves in response to an update in another associated program.

ITU-T International Telecommunications Union - Telecommunications. An organization, with membership including virtually all governments world-wide, that defines and adopts telecommunication standards, regulates frequency distribution, and furthers telecommunication development.

L

LAN Local Area Network. A network connecting workstations, terminals, peripherals, and other devices that reside within a small geographic boundary. LANs typically have a high transfer rate and a low error rate. See also **Ethernet**.

LBO Line Build-Out. T1 circuits need to have their last span lose 15 to 22.3 dB. DTE equipment therefore have selectable output attenuations such as 0, 7.5, or 15 dB of loss at 772 Khtz.

LED Light Emitting Diode. Semiconductor device that changes energy into light when a current is passed through it. LEDs are used as status indicators on all Router devices.

Link A circuit path and all related equipment between a sender and a receiver that is involved in passing a piece of information on the Internet.

LMI Local Management Interface. LMI is a suite of tools, enhancements to Frame Relay, that offers PVC keep-alive functions, global addressing, a multicast mechanism, and a status mechanism.

LOF Loss of Frame. A condition or a signal indicating that the receiving equipment has lost framing data sent by the transmitting device.

Loopback A connection in which a received signal is sent back towards the transmitting device, which waits for its return. Loopbacks are generally used to test Router system WAN and CPE data connections.

LORC Loss Of Receive Clock. Indicates that the transmitting device has lost synchronization with the receiving device. This may result in data slips.

LOS Loss of Signal. A condition indicating that the receiving equipment has lost the incoming signal.

M

MIB Management Information Base. A database of characteristics and parameters that are managed in a network device. MIBS are used to respond to queries by management protocols such as SNMP; they track variables such as error counts and device on/off status.

MFR Multilink Frame Relay. The technique of combining and configuring two or more physical access paths to create a single, high-speed Frame Relay data path.

MLPPP Multi-Link Point-to-Point Protocol. The technique of combining and configuring two or more physical access paths to create a single, high-speed PPP data path.

Modem Modulator/demodulator. A device for converting a digital signal to analog, and vice versa, so that it can be transmitted over phone lines.

MRU Maximum Receivable Unit. The maximum PPP packet size, in bytes, that a particular device can transmit.

MTU Maximum Transmission Unit. The maximum PPP packet size, in bytes, that a particular device can accommodate.

N

NAK Negative Acknowledgment. A control character in ASCII that indicates a packet has arrived with the check digits in error. The packet needs to be retransmitted.

NCM Network Control Module. The component in the Router system that contains its main control circuits. Also known as the motherboard.

NEBS Network Equipment Building Standards. A rigid and extensive set of performance, quality, environmental, and safety requirements for networking equipment, developed by Bellcore.

NNI Network-to-Network Interface. A frame relay and ATM standard protocol that defines how switches establish connections and route signaling requests.

Null-modem cable A cable that allows two computers to talk to each other without a modem. A modified RS-232 cable that has exchanged the modem connector for another PC connector and reversed pins 2 and 3.

O

OOF Out of Frame. A condition that exists when either the DTE or the network detects that either 2 of 4 or 2 of 5 T1 framing bits are missing. OOF conditions that exceed 2.5 seconds generate a local Red Alarm.

OSPF Open Shortest Path First. A routing protocol that minimizes the number of jumps between nodes in a connection.

P

Packet A generic term for a logical grouping of information that consists of a header containing control data, and a payload containing user data. Packets are often referred to as blocks, frames, cells, or datagrams.

PCI Peripheral Component Interconnect. A 32-bit local bus in a PC or Mac that is able to transfer data between the PC's main processor and peripherals at up to 132 megabytes per second.

PDF Portable Document Format. A file format suitable for viewing on any computer terminal. Created and viewed using Adobe Distiller and Acrobat.

PDU Protocol Data Unit. A term used to describe a generic packet containing both a payload and a control-information header.

PING Packet Internet Groper. A test used to determine if a network destination is on-line. An echo request packet is bounced off a destination address and the returned message is analyzed to determine the destination address status.

POP Point of Presence. A physical location where a carrier or ISP has a presence for network access. POPs generally take the form of a switch, router, or concentrator such as the Router system.

Port A physical or electrical interface used to connect to networks or peripherals. Two common types of ports are parallel and serial ports. Also an identifier in transport layer protocols that distinguishes among multiple connections to a single destination.

PPP Point-to-Point Protocol. A protocol that allows both host-to-network and router-to-router connections. PPP is a successor to SLIP and features error detection, data compression, and other features which the older protocol lacked.

PSTN Public Switched Telephone Network. An acronym used to describe the local, long distance, and international phone systems used by the general population.

PVC Permanent Virtual Circuit. A virtual FR circuit between two devices that, once established, transmits all data across the same path. PVCs save bandwidth associated with establishment and tear-down of links that are in almost continuous use.

Q

QoS Quality of Service. A measure of transmission quality, reliability, and availability. QOS is defined in terms of levels; these levels determine what types of applications will run efficiently for a given QOS.

QRW Quasi-Random Waveform. An almost random test signal that places artificial constraints on the number of zeros in a bit sequence. The QRW simulates a live digital signal for testing purposes.

R

RAIS Receive Alarm Indication Signal. Indicates that a device has received an Alarm Indication Signal from an upstream device upstream that has lost its incoming signal. Also known as a yellow alarm in T1 transmission.

RBLU Receive Blue Alarm. A condition indicating that a device has received an all-ones signal indicating an out-of-service condition at the far end. Also known as a AIS.

RED Random Early Detection. RED is used to distribute traffic losses during a router buffer overflow. RED drops packets randomly across many transmitting devices rather than dropping packets from the end of the queue.

REXZ Receive Excessive Errors. A signal has been received by the transmitting device indicating that the destination node is experiencing excessive errors.

RJ-45 A type of cabling with 8 pins.

RLOF Receive Loss of Frame. The transmitting device has received a signal indicating that frame synchronization was lost at the far-end receiving device.

RLOS Receive Loss of Signal. The transmitting device has received a signal indicating that an upstream device has lost the signal.

Routing Updates A message sent by a router to other routers to update network path and cost information. Performed at regular intervals and after changes in network topology.

RPE Receive Parity Error. The transmitting device has received a signal indicating that an upstream device is experiencing parity-bit errors.

RS-232 A long-established standard that describes the physical interface and protocol for relatively low speed serial data communications between computers and related devices.

RSP Response. An answer to an inquiry or the control information sent from a secondary station to a primary station.

S

Service Provider Also called an Internet Service Provider or ISP. An organization that provides access to the internet. A service provider can provide services on the corporate as well as personal level.

SES Severely Errored Second. A second during which the bit error rate exceeds a set limit; during a SES, transmission performance is significantly lowered.

SMTP Simple Mail Transfer Protocol. An internet TCP/IP protocol providing electronic mail services.

SNMP Simple Network Management Protocol. A network management protocol used exclusively in TCP/IP to monitor and control network devices, configurations, and performance information.

Subnet Mask The 32-bit mask that is being used in IP to identify the portion of the IP address that is representing the subnet address.

SW Software. The programs, operating systems, and applications that run, and run on, hardware. A term that distinguishes the instructions (software) from the machines (hardware) that run them.

System Administrator A person responsible for monitoring the operational and administrative functions of a Router system or other network element.

T

T1 A dual twisted-pair cable connection used to transmit digital data at 1.544 Mbps through a network.

TAIS Transmit Alarm Indication Signal. A signal sent to a far-end device to indicate that the signal has been lost at the near-end receiving device.

Tc Time Interval. Variable used to represent an elapsed time.

TCP/IP Transmission Control Protocol/Internet Protocol. An Internet protocol that allows for communication across networks and between computers with varying hardware/software configurations.

Telco Telephone Company. An organization that provides telephone communication services to private and enterprise customers.

Telnet A terminal-remote host protocol that allows a user to sign on to a computer in another city, state, or country and operate it as if he or she was on a hardwired terminal. In Router systems, an access is made via an Ethernet LAN for system management.

TFTP Trivial File Transfer Protocol. Used on a LAN to transport files between a Router system and network host servers.

U

UAS Unavailable Seconds. A count of the number of seconds that a circuit or path is unavailable.

UDP User Datagram Protocol. A connectionless, transport layer protocol that exchanges potentially unreliable, unsequenced, and/or duplicated datagrams with a remote or local user.

UNI User Network Interface. An ATM or Frame Relay Forum specification that define the procedures and protocols between user equipment and either an ATM or Frame Relay Network; UNI is both a physical and functional demarcation point.

UTC Universal Time Coordinated. The official time as kept by all “I” laboratories; “I” laboratories cooperate in the maintenance of Greenwich Mean Time. The Naval Observatory is the US's “I” laboratory.

V

V.35 A high-speed serial interface connecting a Router system to a local router, Frame Relay switch, or another device for WAN access.

VMAP Virtual Multi-megabit Access Path. Technology that enables Router systems to bundle multiple T1 connections into a single path capable of multi-megabit transfer rates.

VRRP Virtual Router Redundancy Protocol. Technology that allows virtual routers to be defined that will default to be the first hop router in the event the actual first hop router fails. In non-VRRP environments, if the first-hop router fails, devices configured for this router lose network/Internet connectivity.

VT-100 Video Terminal - 100. A type of terminal emulation supported by many communications programs; this type of terminal can be used to access a Router system for local management purposes.

W

WAN Wide Area Network. A network that spans a large geographic area and uses high-speed links provided by common carriers such as telephone companies.

X

X.25 A feature rich, widely used standard for packet switching.

INDEX

NOTE: If you are viewing this index with Adobe Acrobat, click the page number of a topic to go to that topic in the guide.

U

V

W

COMMANDS INDEX

NOTE: If you are viewing this index with Adobe Acrobat, click the page number of a topic to go to that topic in the guide.

D

CORPORATE POLICY

Standard Warranty

Hardware

>>company<<> warrants that the Hardware sold hereunder shall be free of defects in workmanship for a period of one (1) year from the Date of Shipment. In the event that >>company<<> receives notice from the Customer during the warranty period that any Hardware does not conform to this warranty, >>company<<>, shall, at its sole option, either repair or replace the non-conforming Hardware. If said notice is received within ninety (90) days of shipment from the >>company<<> factory in 26801 West Agoura Road Calabasas, CA 91301, such replacement shall be "Next Business Day." If said notice is received after ninety (90) days from shipping from >>company<<>, the Customer must ship the defective Hardware to >>company<<> for repair. Such repair shall take no more than ten (10) business days. Cost of shipping to >>company<<> shall be at the Customer's expense. Cost of shipping the repaired unit to the Customer shall be at >>company<<>'s expense. Under the terms of any such warranty, Hardware may be replaced with refurbished or new Hardware at >>company<<>'s option.

For instructions on obtaining Return Material Authorization (RMA), which is required before Hardware under warranty can be returned to Alcatel, refer to Procedures below.

Software

>>company<<> warrants that the media on which the Software is recorded shall be free from defects in material and workmanship under normal use for a period of ninety (90) days from the Date of Shipment. The Customer's sole and exclusive remedy, and >>company<<> sole and exclusive liability, shall be replacement of the media in accordance with this limited warranty. The Customer shall be entitled to free Software upgrades (bug fixes and feature enhancements not listed as separate cost options of the >>company<<> price list) from the >>company<<> website for the first ninety (90) days after shipment from >>company<<>.

Technical Support

Online, phone, and email Technical Support is provided free of charge for a period of ninety (90) days from the Date of Shipment. Technical Support is available online at the >>company<<> website: www.alcatel.com, by phone at: 800 995 2696 (within the United States) or 800 995 4507 (outside the United States), or by email at: support@ind.alcatel.com. Technical Support provides the following services:

- Technical Support will assist the Customer in determining if problems encountered with >>company<<> products are due to errors or limitations in the current >>company<<> operating system and advise the Customer on the availability of Software updates or workarounds.
- Technical Support will provide assistance to the Customer regarding questions concerning the installation of Software updates and configuration questions arising from Software updates.
- Technical Support will work with the Customer to develop and implement appropriate network configurations.

Procedures

A product may only be returned with the prior written approval of >>company<<>. Such approval shall reference a Return Material Authorization (RMA) number issued by authorized >>company<<> Technical Support personnel. To contact Alcatel Technical Support, call 800 995 2696 (within the United States) or 800 995 4507 (outside the United States), or email at support@ind.alcatel.com. Transportation costs, if any, incurred in connection with the return of a defective item to >>company<<> shall be borne by the Customer. Transportation costs incurred in connection with the re-delivery of a repaired or replaced item to the Customer shall be borne by >>company<<>. However, such costs shall be borne by the Customer if >>company<<> reasonably determines that the product is not defective. If >>company<<> determines, in its sole discretion, that the allegedly defective product is not covered by the terms of the warranty provided hereunder, or that a warranty claim is made after the warranty period, the cost of repair by >>company<<>, including all shipping expenses, shall be reimbursed by the Customer. >>company<<>, shall have no liability with respect to data contained in any system returned to >>company<<>.

For complete contact information for Alcatel, see Contacting >>company<<> below.

Exclusions

The foregoing warranties and remedies are for the Customer's exclusive benefit and are non-transferable. Any and all warranties shall be void regarding System components that are damaged or rendered unserviceable by: (1) acts or omissions of non->>company<<> personnel; (2) misuse, theft, vandalism, fire, water, or other peril; (3) alterations of or additions to the System or any element thereof performed by personnel not certified by >>company<<> to perform such alterations and additions or (4) the Customer's failure to meet environmental specifications.

Non->>company<<> Products

In circumstances where a product not manufactured or created by >>company<<> is sold by >>company<<> hereunder to complete an order, the Customer's sole remedy shall be pursuant to the original manufacturer's/licensor's warranty to the Customer, to the extent permitted by the original manufacturer/licensor.

Disclaimer of Warranty

The limited warranties referred to in paragraphs above shall be in lieu of all other warranties whether expressed, implied, statutory, or otherwise. >>company<<> specifically disclaims any implied warranties of merchantability or fitness for a particular purpose.

Limitation of Liability

>>company<<> and its suppliers exclude themselves from any liability for any lost revenue or profit, loss of business, loss of information or data, or for special, indirect, consequential, incidental, or punitive damages of any kind caused out of or in connection with the sale, installation, maintenance, use, performance, failure, or interruption of its products, even if >>company<<> and its authorized resellers have been advised of the possibility of such damages. In no event shall >>company<<> or its supplier's total liability to the Customer, whether in contract negligence, strict liability, tort or otherwise, exceed the price paid by the Customer. The foregoing limitations shall apply even if any remedy provided herein shall fail its essential purpose.

Equipment Malfunction

In the event this equipment should fail to operate properly, disconnect the unit from the telephone line and contact >>company<<> Technical Support at the address and phone number listed in below. DO NOT DISASSEMBLE THIS EQUIPMENT. This equipment does not contain any user-serviceable components. If the equipment is causing harm to the network, the telephone company may request that you disconnect this equipment from the telephone network until the problem is resolved.

Contacting >>company<<>

Telephone	818 880 3500
Fax	818 880 3505
Mail	26801 West Agoura Road Calabasas, CA 91301
Sales	818 880 3500
Support	800 995 2696 (US) 800 995 4507 (International)
Email	support@ind.alcatel.com
Internet	http://eservice.ind.alcatel.com