Command Line Interface Commands Reference

Firmware Version 8.7.4

Motorola Netopia® ENT-Series Routers

MOTOROLA

Copyright

Motorola, Inc.
1303 East Algonquin Road
Schaumburg, Illinois 60196
USA

Firmware Version 8.7.4

Part Number

Motorola part number 6160028-00-24

Contents

Chapter 1

Introduction

This *Command Line Interface Commands Reference* contains information on the syntax and use of the Command Line Interface for the Motorola Netopia® router family. It provides information required to configure the router firmware and troubleshoot problems using the Command Line Interface.

This document is intended for small office, home office, and remote office users, and other networking professionals who administer networks using Motorola Netopia® routers.

Note: Restrictions among firmware versions are noted in the body of this document. Where no firmware version is noted, the commands given are supported on all platforms.

New Commands in Firmware Version 8.7.4

Firmware Version 8.7.4 adds new and revised commands for the following:

- IP multicast to layer 2 unicast mapping. See "IGMP Configuration Commands" on page 2-70.
- Change backup timer from minutes to seconds. See "Backup Configuration Commands" on page 2-115.
- Support for router-generated packets with source address outside of local member range for IPSec force all tunnels. See "Unprotected Services Configuration Commands" on page 2-68.
- Enhanced VLAN Support and configuration changes. See "Virtual LAN (VLAN) configuration commands" on page 2-36.
- DHCP Filterset, Option Groups, DHCP Generic Options support. See "DHCP Gen-Options, Option Groups, and Option Filtersets Commands" on page 2-77.

Syntax Notation

The command descriptions use formatted text to indicate various attributes of each command. The syntax is as follows:

- Required keywords and commands that must be typed literally are in **boldface**.
- Optional elements are enclosed in square brackets "[]".
- Mutually exclusive elements are contained in braces "{}" and separated by vertical bars "|".
- Arguments for which you supply values are in *italics*.
- Examples of commands you type and the results of those commands are in the `courier` typeface.
- An element that may be repeated one or more times is followed by a superscripted plus sign "$^{+}$".
- An element that may be repeated zero or more times is followed by a superscripted asterisk "*".

Interface Naming Conventions

A number of commands described in this document require you to identify the router interface to be affected by the command. This requires specifying both an interface type (denoted *intf-type*) and an interface index (denoted *id*).

The *intf-type* argument may be replaced with one of the following keywords:

`adsl | aux | dds | ethernet | isdn | sdsl | t1 | wan | serial`

If a command is not specific to a particular WAN interface type, the *intf-type* **wan** may be specified; otherwise, the more specific *intf-type* must be specified.

Note: For IDSL interfaces, use the keyword **isdn**.

The *id* argument can be replaced with 0, 1, or 2, as follows:

- **0** means the motherboard
- **1** means the WAN 1 slot
- **2** means the WAN 2 slot

So, for example, the ethernet hublet is identified as “interface ethernet 0”. In some contexts, only a WAN interface may be specified, in which case the command syntax will specify *wan-id* instead of the more general *id*. The *wan-id* argument can be replaced by either **1** (the WAN 1 slot) or **2** (the WAN 2 slot) on R-Series equipment.

Security (Configuration Access)

If the device is password-protected, the device requires you to enter a name and password before you can access the menu-based or command line console interface. See the section “Configuration Access Commands” on page 2-3.

Entering and Editing Commands

The device’s console user interface comes up in Menu mode by default. In this mode you use the arrow, Escape, and Return/Enter keys to navigate through a series of screens. To invoke the command line at any time, type **Control-N**. The console will erase the window, and you will be presented with a **#** prompt. To return to Menu mode type Control-N again.

The following table provides a description of keys that can be used when entering and editing commands. Control indicates the Control key, which must be pressed simultaneously with the associated letter key. Escape indicates the Escape key, which must be pressed and released first, followed by its associated letter key. Keys are not case-sensitive.

Command Editing Keys and Functions	
Key	**Function**
Control-A	Moves the cursor to the beginning of the command line.
Control-E	Moves the cursor to the end of the command line.
Control-K	Deletes all characters from the cursor to the end of the command line.
Control-N	Invokes the command line interface from the menu console. Invokes the menu console from the command line interface.
Control-U	Deletes all characters from the cursor back to the beginning of the command line.
Control-W	Deletes the word to the left of the cursor.
Escape B	Moves the cursor back one word.
Escape D	Deletes from the cursor to the end of the word.
Escape F	Moves the cursor forward one word.

Online Help

Beginning with Firmware Version 8.6, online help is available to prompt you when entering commands. If you enter a partial or incorrect command, the help facility displays prompts to alert you to the correct syntax for the command. The help facility offers expected keywords from which to select, and an explanation of error messages.

Example:

If you enter the partial command **show snmp**, the help facilty will prompt you with the appropriate keywords until the command is successfully entered.

```
#show snmp
; error 103: incomplete command
; acceptable next keywords:
;  authentication
;  community
;  heartbeat-interval
;  notify
;  system
;  trap
#show snmp authentication
; error 103: incomplete command
; next keyword must be:
;  traps
#show snmp authentication traps
; error 103: incomplete command
; next keyword must be:
;  enable
#show snmp authentication traps enable
snmp authentication traps enable no
```

Chapter 2

Motorola Netopia® Router CLI Commands

This chapter describes the syntax of the supported command set of the Motorola Netopia® R-series, 4000-series, and 3000 Enterprise-series Router families.

Configuration Access Commands

Configuration Access Commands
date ***xx/yy/zz***
show date
exit
preferences changes immediate { yes | no }
show preferences changes immediate
no preferences changes immediate
preferences check vci { yes | no }
preferences console default { **menu** | **cli** }
show preferences console default
preferences console timeout *seconds*
no preferences console timeout
show preferences console timeout
preferences date format { **mm/dd/yy** | **dd/mm/yy** | **yy/mm/dd** }
show preferences date format
preferences output format { **terse** | **verbose** }
show preferences output format
preferences output mask { **bits** | **dotted-quad** }
show preferences output mask
preferences time format { **am-pm** | **24-hour** }
show preferences time format
security password
no security password
snmp authentication traps enable [**yes** | **no**]
no snmp authentication traps enable
show snmp authentication traps enable
snmp community { **ro** | **read-only** | **rw** | **read-write** } *string*
no snmp community [**ro** | **read-only** | **rw** | **read-write**] [*string*]

Configuration Access Commands (cont. 1)
snmp heartbeat-interval *interval* **show snmp heartbeat-interval** **no snmp heartbeat-interval** **snmp notify type** [**v1-trap** | **v2-trap** | **inform**] **snmp system contact** *string* **show snmp system contact** **no snmp system contact** **snmp system location** *string* **show snmp system location** **no snmp system location** **snmp system name** *string* **show snmp system name** **no snmp system name** **snmp system trap source address** [**lan** | **wan**] **system syslog enable** { **yes** | **no** } **no system syslog enable** **show system syslog enable** **system syslog host-name** *hostname* **no system syslog host-name** **show system syslog host-name** **system syslog facility** *facility* **show system syslog facility** **system syslog log-violations** { **yes** | **no** } **no system syslog log-violations** **show system syslog log-violations** **system syslog log-accepts** { **yes** | **no** } **no system syslog log-accepts** **show system syslog log-accepts**

Configuration Access Commands (cont. 2)

system syslog log-attempts { **yes** | **no** }
no system syslog log-attempts
show system syslog log-attempts

telnet { *hostname* | *ip-addr* } [**port** *value*] [**source** *ip_addr*]
show telnet sessions

telnet suspend [**a... z**]
show telnet suspend

telnet resume [**1... 6**]

telnet terminate [**1... 6**]

telnet server port [*port number*]
show telnet server port

time *hh:mm* [**am** | **pm**]
show time

user *name password*
no user *name* [*password*]

MAC Address Security Commands

security mac-auth mode [**disabled** | **allow-list** | **deny-list**]
show security mac-auth mode

security mac-auth wireless-only [**yes** | **no**]
show security mac-auth wireless-only

security mac-auth mac-deny *MAC-addr*
show security mac-deny

security mac-auth mac-allow *MAC-addr*
show security mac-auth mac-allow

The **preferences** command allows you to customize certain aspects of the command line interface. Preference settings persist across restarts, and are specific to the user name, if any, you used to authenticate yourself before issuing the **preferences** command. If no users are defined, no authentication is required, and preference settings are global.

---

date *xx/yy/zz*
show date

These commands allow you to set or display the current date for the router's system clock.

---

exit

The **exit** command terminates your current console session. If you are connected via telnet or a modem, the connection will be closed. If you are logged in via the serial console, you will return to the command line or menu-based console based on your default console setting. (See the **preferences console default** command on page 2-6.) In either case, you will be prompted either with a login prompt (if one or more users are defined), or the initial prompt for the selected console interface (if no users are defined).

---

preferences changes immediate { **yes** | **no** }
show preferences changes immediate
no preferences changes immediate

These commands allow you to specify whether or not WAN configuration changes will take effect immediately. When you specify **no**, any changes you make to the WAN configuration (except NAT) will not take effect until the router is reset.

Note: The router will reboot immediately when the value of the **changes immediate** preference item changes. No warning is given.

---

preferences check vci { **yes** | **no** }

Note: This command is supported beginning with firmware version 8.2.

This command allows you to set the VCI to any value in the CLI, menu, or SNMP. The ability to set a VCI to 0 – 31 is not normally permitted. This command allows you to override this check and allow a VCI of value 0 – 31.

---

preferences console default { **menu** | **cli** }
show preferences console default

The **preferences console default** command specifies the console interface that will be presented to the user on subsequent logins. When set to **menu** (the default), the user will be presented with the menu-based console interface on subsequent logins. When set to **cli**, the user will be presented with the command line console interface on subsequent logins. If the **preferences console default** command is issued and there are no users defined, the setting will determine the console interface that will be presented to all newly established console sessions (via either the serial console port or via telnet).

---

preferences console timeout *seconds*
no preferences console timeout
show preferences console timeout

These commands control the command-line and menu-based console auto logout. Note that the **no preferences console timeout** command sets the timeout to zero, which disables the timeout.

The command:

```
no preferences console timeout
```

is equivalent to:

```
preferences console timeout 0
```

Example:

```
preferences console timeout 300
```

---

preferences date format { **mm/dd/yy** | **dd/mm/yy** | **yy/mm/dd** }
show preferences date format

These commands allow you to set or display your date formatting preferences for the router's system clock.

---

preferences output format { **terse** | **verbose** }
show preferences output format

The **preferences output format** command affects the format of the output from show commands. When set to **verbose** (the default), the output from **show** commands is formatted as a valid command line interface command that could be entered at a command prompt. When set to **terse**, the output from show commands is *not* formatted as a valid command line interface command that could be entered at a command prompt, but rather includes only the value of the requested attribute. The **terse** mode may be more useful if the output will be processed by a computer rather than a human being.

Example:

```
#preferences output format verbose
#show interface ethernet 0 ip address
interface ethernet 0 ip address 192.168.1.1/24
#preferences output format terse
#show interface ethernet 0 ip address
192.168.1.1/24
```

---

preferences output mask { **bits** | **dotted-quad** }
show preferences output mask

The **preferences output mask** command affects the format of the output from those **show** commands that display an IP address together with a subnet mask. When set to **bits** (the default), the IP address and subnet mask are output in prefix notation – i.e., an IP address in dotted-quad notation followed by a slash followed by the number of consecutive ones-bits in the subnet mask – whereas when set to **dotted-quad**, the IP address and subnet mask are output as two consecutive dotted-quads.

Example:

```
#preferences output mask bits
#show interface ethernet 0 ip address
interface ethernet 0 ip address 192.168.1.1/24
#preferences output mask dotted-quad
#show interface ethernet 0 ip address
interface ethernet 0 ip address 192.168.1.1 255.255.255.0
```

---

preferences time format { **am-pm** | **24-hour** }
show preferences time format

These commands allow you to set or display your time formatting preferences for the router's system clock.

---

security password
Enter old password: *old password*
Enter new password: *new password*
Re-enter password: *new password*

no security password
Enter old password:*old password*

These commands let you set and delete the **Security Options** screen password. After you enter the command the console prompts you for the existing password if you have one, then it prompts you to enter and re-enter a new password (eleven characters maximum). The **no** command will prompt you for a password if there was one, and will then delete that password.

---

snmp authentication traps enable [**yes** | **no**]
no snmp authentication traps enable
show snmp authentication traps enable

These commands allow you to enable, disable, or show the status of SNMP authentication traps.

---

snmp community { **ro** | **read-only** | **rw** | **read-write** } *string*
no snmp community [**ro** | **read-only** | **rw** | **read-write**] [*string*]

These commands allow you to add or delete the SNMP community Read-Only and Read-Write strings.

snmp heartbeat-interval *interval*
show snmp heartbeat-interval
no snmp heartbeat-interval

Note: These commands are supported beginning with firmware version 8.2.

These commands allow you to set, show, or delete the SNMP heartbeat interval. A single configuration item governs heartbeat traps, the time interval between traps. Permitted values are 0 – 65535 minutes. A value of zero, the default, means the trap is disabled. This value can be configured by the CLI and SNMP. When the interval value is set to a positive number, a trap is sent immediately and the new (or same) interval value takes effect.

snmp notify type [**v1-trap** | **v2-trap** | **inform**]

Note: This command is supported beginning with firmware version 8.4.2.

This command allows you to set the type of SNMP traps that the system will generate: **v1**, **v2**(c), or, beginning with Firmware Version 8.4.2, **inform**.

snmp system contact *string*
show snmp system contact
no snmp system contact

These commands set, display, or clear the router's SNMP system contact (sysContact) string.

snmp system location *string*
show snmp system location
no snmp system location

These commands set, display, or clear the router's SNMP system location (sysLocation) string.

snmp system name *string*
show snmp system name
no snmp system name

These commands set, display, or clear the router's SNMP system name (sysName) string.

snmp system trap source address [**lan** | **wan**]

Note: This command is supported beginning with firmware version 8.5.

This command allows you to specify whether the source address for SNMP traps should be on the LAN or the WAN. When this parameter is set to **lan**, all SNMP v2 and inform traps use the source IP address of the primary LAN interface. Otherwise, the IP address of the WAN interface is used.

system syslog enable { **yes** | **no** }
no system syslog enable
show system syslog enable

Note: These commands are supported beginning with Firmware Version 8.2.

These commands allow you to enable, disable, or show the status of logging of system events for reporting by a Syslog client. By default, all events are logged in the event history. By using the syslog commands that follow to set each event descriptor to either **yes** or **no**, you can determine which ones are logged and which are ignored.

---

system syslog host-name *hostname*
no system syslog host-name
show system syslog host-name

Note: These commands are supported beginning with Firmware Version 8.2.

These commands allow you to specify, disable, or show the status of the syslog server's address either in dotted decimal format or as a DNS name up to 64 characters.

---

system syslog facility *facility*
show system syslog facility

Note: These commands are supported beginning with Firmware Version 8.2.

These commands allow you to specify or show the UNIX syslog Facility. *facility* values may be "local0" through "local7".

---

system syslog log-violations { **yes** | **no** }
no system syslog log-violations
show system syslog log-violations

Note: These commands are supported beginning with Firmware Version 8.2.

These commands allow you to enable, disable, or show whether violations are logged or ignored.

---

system syslog log-accepts { **yes** | **no** }
no system syslog log-accepts
show system syslog log-accepts

Note: These commands are supported beginning with Firmware Version 8.2.

These commands allow you to enable, disable, or show whether acceptances are logged or ignored.

---

system syslog log-attempts { **yes** | **no** }
no system syslog log-attempts
show system syslog log-attempts

Note: These commands are supported beginning with Firmware Version 8.2.

These commands allow you to enable, disable, or show whether connection attempts are logged or ignored.

---

telnet { *hostname* | *ip-addr* } [**port** *value*] [**source** *ip_addr*]
show telnet sessions

Note: These commands are supported beginning with Firmware Version 8.7.

These commands allow you to initiate or show up to six telnet sessions from the command line without returning to the console menu interface. Using the command line, you can resume sessions started by the console menu and vice versa.

Example:

```
#show telnet sessions
#1 192.168.1.253
#2 192.168.1.91
#3 10.8.200.16
#4 no active session
#5 no active session
#6 no active session
```

---

telnet suspend [**a... z**]
show telnet suspend
telnet resume [**1... 6**]
telnet terminate [**1... 6**]

Note: These commands are supported beginning with Firmware Version 8.7.

These commands allow you to suspend telnet sessions indicated by alphabetic letter, **a** through **z**, from the command line.

Telnet sessions specified by number, **1** through **6**, may be resumed or terminated.

The **show** command displays telnet sessions that have been previously suspended using the **suspend** command.

---

telnet server port [*port number*]
show telnet server port

These commands allow you to set or display the TCP port on which the router is currently listening for incoming telnet management sessions. If you change the port number, the router will immediately stop accepting new sessions at the old port number, and only accept incoming sessions on the new port number. All sessions currently connected to the old port number will remain connected. Allowed values for port number are 1 - 65535, except for 80 and 1723.

---

time *hh:mm* [**am** | **pm**]
show time

These commands allow you to set or display the current time for the router's system clock.

MAC Address Security Commands

Note: These commands are supported beginning with firmware version 8.5.

security mac-auth mode [**disabled** | **allow-list** | **deny-list**]
show security mac-auth mode

These commands allow you to configure or display the global MAC authentication mode. If set to **allow-list**, all non-matching unicasts will be dropped. If set to **deny-list**, all matching unicasts will be dropped.

security mac-auth wireless-only [**yes** | **no**]
show security mac-auth wireless-only

These commands allow you to restrict or display the restricted status of MAC address authentication. If set to **yes**, the MAC authentication applies only to the wireless interface, on models so equipped. If set to **no**, packets received at all interfaces on the LAN are subject to the MAC filtering table.

security mac-auth mac-deny *MAC-addr*
show security mac-deny

These commands allow you to specify or display the MAC address for hosts on the wired or wireless LAN (if so restricted) whose source or destination MAC address will cause the router to filter their packets.

security mac-auth mac-allow *MAC-addr*
show security mac-auth mac-allow

These commands allow you to specify or display the MAC address for hosts on the wired or wireless LAN (if so restricted) whose source or destination MAC address will cause the router to pass their packets.

Note: Wireless MAC authentication commands are also supported. See "Wireless Configuration Commands" on page 2-86.

System Heartbeat Configuration Commands

Note: The commands in this section are supported beginning with firmware version 8.5.

System Heartbeat Configuration Commands
heartbeat enable { **yes** | **no** } **show heartbeat enable**
heartbeat protocol { **udp** | **tcp** } **show heartbeat protocol**
heartbeat client-port *port* **show heartbeat client-port**
heartbeat interval *time* (in seconds) **show heartbeat interval time**
heartbeat count *count* **show heartbeat count**
heartbeat sleep-time *time* (in seconds) **show heartbeat sleep-time**
heartbeat server port *port* **show heartbeat server port**
heartbeat server address *address* **show heartbeat server address**
heartbeat server url *url* **show heartbeat server url**
heartbeat interval contact-email *email_address* **show heartbeat interval contact-email**
heartbeat interval location *location* **show heartbeat interval location**
reset heartbeat

---

heartbeat enable { **yes** | **no** }
show heartbeat enable

These commands allow you to enable, disable, or show the status of the system heartbeat.

Once a unit is configured and restarted, the WAN link is up and the WAN IP address is established, the heartbeat will begin executing and sending its payloads (or establishing its connection in the case of TCP). A special case is when the ip-server address is on the LAN. In this case, the payloads will be routed to the LAN side address, but only after the WAN link and WAN IP addresses have been established.

If, at any time during the heartbeat sequence, the link state changes – which means, for example, that there is a layer 1 disconnect or a change in the IP layer parameters from a DHCP acquisition or a renegotiated PPP session – the sequence will restart. You can also restart the sequence manually. In addition, in TCP mode once the connection has been established, the sequence will be restarted any time the remote server closes it.

---

heartbeat protocol { **udp** | **tcp** }
show heartbeat protocol

These commands allow you to specify or show the protocol to be used for the system heartbeat, **udp** or **tcp**.

The heartbeat is a state machine:

- If you select **udp**, there are no connections to the server. If the server address is known, it simply sends the payloads in UDP.
- If you select **tcp**, it tries to connect to the server address, and keeps trying to connect for 20 attempts at thirty-second intervals. If a connection is not established, it sleeps for a minimum of either 30 minutes, or whatever is programmed as the **sleep-time**. See below.

---

heartbeat client-port *port*
show heartbeat client-port

These commands allow you to specify or show the client port to be used for the system heartbeat.

---

heartbeat interval *time* (in seconds)
show heartbeat interval

These commands allow you to specify or show the heartbeat interval. in seconds.

---

heartbeat count *count*
show heartbeat count

These commands allow you to specify or show the heartbeat count within the specified **interval**.

---

heartbeat sleep-time *time* (in seconds)
show heartbeat sleep-time

These commands allow you to specify or show the heartbeat sleep time, in seconds, during which the system will wait before retrying a failed connection attempt, if **tcp** is the specified protocol.

---

heartbeat server port *port*
show heartbeat server port

These commands allow you to specify or show the heartbeat server port number.

heartbeat server address *address*
show heartbeat server address

These commands allow you to specify or show the heartbeat server IP address. Beginning with Firmware Version 8.5.1, the address can also be a DNS name of up to 63 characters.

heartbeat server url *url*
show heartbeat server url

These commands allow you to specify or show a heartbeat server URL.

heartbeat interval contact-email *email_address*
show heartbeat interval contact-email

These commands allow you to specify or show an email address to be placed into the heartbeat Xml payload.

heartbeat interval location *location*
show heartbeat interval location

These commands allow you to specify or show a location to be placed into the heartbeat Xml payload.

reset heartbeat

This command allows you to restart the heartbeat sequence.

Tiered Configuration Access Commands

Tiered Configuration Access Commands
superuser *name password* **show superuser** **no superuser** **user** *name password* [{ **wan** | **lan** | **cp** | **nat** | **pvc** | **global** | **subnet** | **voice** | **no-web** | **no-telnet** }*] **show user** **no user** *name*

superuser *name password*
show superuser
no superuser

These commands allow you to create, show, or delete a Superuser. You can only configure a Superuser if no authorized users exist. There can be but one Superuser. The Superuser can change any attributes of any user, including itself. However, even the Superuser cannot see what the password for a user is – the **show** command will display 5 asterisks regardless of its actual length.

user *name password* [{ **wan** | **lan** | **cp** | **nat** | **pvc** | **global** | **subnet** | **no-telnet** }*]
show user
no user *name*

These commands allow a Superuser to create, show, or delete a user and his/her access privileges. A user can change only his/her own password, and cannot change their access privileges. If a Superuser creates a new user, this user inherits the privileges of the first non-Superuser, or has the default access privileges of **lan** | **subnet** | **nat** | **cp** | **global** if there is no non-Superuser configured.

Permissible modifiers are:

wan	WAN interface(s) configuration
lan	LAN (Ethernet *id*) interface configuration
cp	connection profile (and default profile) configuration
nat	Network Address Translation configuration. This includes the ability to configure NAT attributes in connection profiles.
pvc	ATM PVC and Frame DLCI configuration
global	other parameters, such as console preferences. This includes ping and traceroute functionality.
subnet	LAN (Ethernet 0) interface ip subnet configuration
no-telnet	Prevents Telnet access.

Interface Configuration Commands

Additional LAN configuration command

Note: Beginning with Firmware Version 8.4.2, the firmware includes support for creating additional logical local area networks. When used in combination with VLANs (see "Virtual LAN (VLAN) configuration commands" on page 2-36), you can maintain separate functional end-to-end networks to support such services as voice-over-IP, point-of-sale applications, or audio and video services.

Multiple logical IP LAN support allows you to create additional IP routed LAN interfaces (ALANs). You can add, edit, or delete Additional LANs similarly to Connection Profiles on the WAN connection. You then associate physical or logical Ethernet-encapsulated interfaces, such as wired Ethernet ports, wireless SSIDs, and ATM RFC 1483 bridged VCs by attaching the ALAN to a VLAN containing these interfaces.

The additional LAN IP routed interfaces duplicate all the same parameters that apply to the primary LAN interface, such as DHCP servers, filtersets, multicast forwarding, and RIP. You can configure up to six ALANs.

interface ethernet *id* [**yes** | **no**]

This command allows you to create or delete an additional LAN (ALAN) of id *id*. If you create an ALAN, you must provision it with the same parameters that apply to the primary LAN.

Example:

```
interface ethernet 2 yes
interface ethernet 2 tag "Telecommuter"
interface ethernet 2 enable yes
interface ethernet 2 ip address 3.0.0.1/8
interface ethernet 2 ip rip receive both
interface ethernet 2 ip rip transmit no
interface ethernet 2 ip multicast-fwd no
interface ethernet 2 address-serve enable no
interface ethernet 2 address-serve clients none
interface ethernet 2 address-serve dhcp lease-time 1
interface ethernet 2 address-serve gateway default 3.0.0.1/8
interface ethernet 2 address-serve mode server
interface ethernet 2 address-serve netbios mode enable no
interface ethernet 2 address-serve netbios mode type b-node
interface ethernet 2 address-serve netbios scope enable no
interface ethernet 2 address-serve netbios name-server enable no
interface ethernet 2 address-serve netbios name-server address 0.0.0.0
interface ethernet 2 address-serve range 3.0.0.100 3.0.0.199
interface ethernet 2 mac address 00:00:c5:fa:dd:04
```

Ethernet Interface configuration commands

Ethernet Interface Configuration Commands
interface ethernet *id* **ip address** [{ *ip-addr/ mask-bits* | *ip-addr mask* | **secondary** }] **no interface ethernet** *id* **ip address** [{ *ip-addr/mask-bits* | *ip-addr mask* | **secondary** }] **show interface ethernet** *id* **ip address**
interface ethernet *id* **ip dhcp client mode** { **standard** | **copper-mountain** | **cmn** } **show interface ethernet** *id* **ip dhcp client mode**
interface ethernet *id* **ip multicast-fwd** { **yes** | **no** } **no interface ethernet** *id* **ip multicast-fwd** **show interface ethernet** *id* **ip multicast-fwd**
interface ethernet *id* **ip igmp-version** { **v1** | **v2 l v3** } **show interface ethernet** *id* **ip igmp-version**
interface ethernet *id* **mac address** { *MAC-address* | **default** } **show interface ethernet** *id* **mac address**
interface ethernet *id* **mode** { **autonegotiate** | **100full** | **100half** | **10full** | **10half** | **100full-fixed** | **100half-fixed** | **10full-fixed** | **10half-fixed** } **show interface ethernet** *id* **mode**
interface ethernet *id* **ip nat enable** { **yes** | **no** } **no interface ethernet** *id* **ip nat enable** **show interface ethernet** *id* **ip nat enable**
interface ethernet *id* **ip nat map-list** *list-tag* **no interface ethernet** *id* **ip nat map-list** **show interface ethernet** *id* **ip nat map-list**
interface ethernet *wan-id* **ip nat passthrough enable** { **yes** | **no** } **no interface ethernet** *id* **ip nat passthrough enable** **show interface ethernet** *id* **ip nat passthrough enable**
interface ethernet *wan-id* **ip nat passthrough dhcp enable** { **yes** | **no** } **no interface ethernet** *wan-id* **ip nat passthrough dhcp enable** **show interface ethernet** *wan-id* **ip nat passthrough dhcp enable**

Ethernet Interface Configuration Commands (continued)

interface ethernet *wan-id* **ip nat passthrough dhcp mac-address** { *mac-address* }
show interface ethernet *wan-id* **ip nat passthrough dhcp mac-address**

interface ethernet *id* **ip nat server-list** *list-tag*
no interface ethernet *id* **ip nat server-list**
show interface ethernet *id* **ip nat server-list**

interface ethernet *id* **ip netbios proxy enable** { **yes** | **no** }
no interface ethernet *id* **ip netbios proxy enable**
show interface ethernet *id* **ip netbios proxy enable**

interface ethernet *id* **pppoe enable** { **yes** | **no** }
no interface ethernet *id* **pppoe enable**
show interface ethernet *id* **pppoe enable**

show interface ethernet *id* **statistics**
show interface ethernet *id* **stats**

interface ethernet *id* **ip filterset** *fs-id*
no interface ethernet *id* **ip filterset**
show interface ethernet *id* **ip filterset**

Ethernet Interface RIP Configuration Commands
interface ethernet *id* **ip rip exclude-wan-routes** **no interface ethernet** *id* **ip rip exclude-wan-routes** **show interface ethernet** *id* **ip rip exclude-wan-routes**
interface ethernet *id* **ip rip receive** { **no** | **v1** | **v2** | **both** | **v2-md5** } **no interface ethernet** *id* **ip rip receive** **show interface ethernet** *id* **ip rip receive**
interface ethernet *id* **ip rip transmit** { **no** | **v1** | **v2broadcast** | **v2multicast** | **v2broadcast-md5** | **v2multicast-md5** } **no interface ethernet** *id* **ip rip transmit** **show interface ethernet** *id* **ip rip transmit**
interface ethernet *id* **ip rip auth key** *id* **no interface ethernet** *id* **ip rip auth key** *id* **show config interface ethernet** *id* **ip rip auth key**
interface ethernet *id* **ip rip auth key** *id* **start date** *date* **show interface ethernet** *id* **ip rip auth key** *id* **start date**
interface ethernet *id* **ip rip auth key** *id* **start time** *time* **show interface ethernet** *id* **ip rip auth key** *id* **start time**
interface ethernet *id* **ip rip auth key** *id* **end date** *date* **show interface ethernet** *id* **ip rip auth key** *id* **end date**
interface ethernet *id* **ip rip auth key** *id* **end time** *time* **show interface ethernet** *id* **ip rip auth key** *id* **end time**
interface ethernet *id* **rip auth key** *id* **end time mode** { **infinite** | **date** } **show interface ethernet** *id* **ip rip auth key** *id* **end time mode**
interface ethernet *id* **rip auth key** *id* **key** <*string*>

Ethernet Interface IP Address Serving Commands

interface ethernet *id* **address-serve clients** { **any** | **none** | { **bootp** | **dhcp** | **macip** | **wan** }$^{+}$ }
no interface ethernet *id* **address-serve clients** { **any** | { **bootp** | **dhcp** | **macip** | **wan** }$^{+}$ }
show interface ethernet *id* **address-serve clients**

interface ethernet *id* **address-serve dhcp enable { yes | no }**
no interface ethernet *id* **address-serve dhcp enable**
show interface ethernet *id* **address-serve dhcp enable**

interface ethernet *id* **address-serve dhcp dns [1 | 2]** *ip-addr*

interface ethernet *id* **address-serve dhcp lease-time** *hours*
show interface ethernet *id* **address-serve dhcp lease-time**

interface ethernet *id* **address-serve dhcp option 150 address** *www.xxx.yyy.zzz*
no interface ethernet *id* **address-serve dhcp option 150 address** *www.xxx.yyy.zzz*
show interface ethernet *id* **address-serve dhcp option 150 address**

show interface ethernet *id* **address-serve dhcp addresses**

show interface ethernet *id* **ip dhcp client status**

interface ethernet *id* **ip dhcp client** [**renew** | **release**]

interface ethernet *id* **address-serve dhcp next-server** *ip-addr*

interface ethernet *id* **address-serve gateway** { *gw-ip-addr* | **default** { *ip-addr/mask-bits* | *ip-addr mask* } }
show interface ethernet *id* **address-serve gateway**

interface ethernet *id* **address-serve helper** *ip-addr*
no interface ethernet *id* **address-serve helper** [*ip-addr*]
show interface ethernet *id* **address-serve helper**

interface ethernet *id* **address-serve mode** { **relay** | **server** }
show interface ethernet *id* **address-serve mode**

interface ethernet *id* **address-serve range** { **auto** | *from-addr to-addr* }
no interface ethernet *id* **address-serve range** *from-addr to-addr*
show interface ethernet *id* **address-serve range**

interface ethernet *id* **address-serve** { **no** | **off** | **on** | **yes** }
no interface ethernet *id* **address-serve**
show interface ethernet *id* **address-serve**

Ethernet Interface Stateful Inspection Commands

Ethernet Interface Stateful Inspection Commands
interface ethernet *id* **ip state-insp enable** { **yes** | **no** | **on** | **off** } **no interface ethernet** *id* **ip state-insp** **show interface ethernet** *id* **ip state-insp enable**
interface ethernet *id* **ip state-insp router-access** { **yes** | **no** | **on** | **off** } **no interface ethernet** *id* **ip state-insp router-access** **show interface ethernet** *id* **ip state-insp router-access**
interface ethernet *id* **ip state-insp tcp-seq-diff** *diff* **show interface ethernet** *id* **ip state-insp tcp-seq-diff**
interface ethernet *id* **ip state-insp deny-frag** { **yes** | **no** | **on** | **off** } **no interface ethernet** *id* **ip state-insp deny-frag** **show interface ethernet** *id* **ip state-insp deny-frag**
interface ethernet *id* **ip state-insp xposed-list** *xposed-list_name* **no interface ethernet** *id* **ip state-insp xposed-list** **show interface ethernet** *id* **ip state-insp xposed-list**

Ethernet Interface Static Client Address Translation Commands
interface ethernet *lan_interface_id* **scat enable** [**yes** | **no**] **show interface ethernet** *lan_interface_id* **scat enable**

Ethernet Interface VRRP Commands

interface ethernet *id* **ip vrrp vrouter** *id* **vrid** *vrid*
show interface ethernet *id* **ip vrrp vrouter** *id* **vrid**

interface ethernet *id* **ip vrrp vrouter** *id* **vrip** *ip-addr*
show interface ethernet *id* **ip vrrp vrouter** *id* **vrip**

interface ethernet *id* **ip vrrp vrouter** *id* **priority** [**1... 255**]
show interface ethernet *id* **ip vrrp vrouter** *id* **priority**

interface ethernet *id* **ip vrrp vrouter** *id* **adv-intvl** [**1... 255**]
show interface ethernet *id* **ip vrrp vrouter** *id* **adv-intvl**

interface ethernet *id* **ip vrrp vrouter** *id* **preempt-mode enable** [**no** | **yes** | **on** | **off**]
show interface ethernet *id* **ip vrrp vrouter** *id* **preempt-mode enable**

interface ethernet *id* **ip vrrp vrouter** *id* **enable** [**no** | **yes** | **on** | **off**]
show interface ethernet *id* **ip vrrp vrouter** *id* **enable**

no interface ethernet *id* **ip vrrp vrouter** *id*

show interface ethernet *id* **ip vrrp wan-monitor enable** [{ **yes** | **no** | **on** | **off** }]
show interface ethernet *id* **ip vrrp wan-monitor enable**

interface ethernet *id* **ip vrrp master-dhcp-srv enable** [{ **yes** | **no** | **on** | **off** }]
show interface ethernet *id* **ip vrrp master-dhcp-srv enable**

interface ethernet *id* **ip vrrp dhcp-gateway** *ip-addr*
show interface ethernet *id* **ip vrrp dhcp-gateway**

interface ethernet *id* **ip address** [{ *ip-addr/ mask-bits* | *ip-addr mask* | **secondary** }]
no interface ethernet *id* **ip address** [{ *ip-addr/mask-bits* | *ip-addr mask* | **secondary** }]
show interface ethernet *id* **ip address**

These commands allow you to set, delete, or show the IP subnet(s) of an Ethernet interface. If the keyword **secondary** is specified in the first command, the subnet is appended to the list of subnets (assuming that all of the allowed subnets have not yet been configured– the router supports up to eight). If the keyword **secondary** is not specified, the primary subnet configuration is replaced with the specified values. The mask may be specified either as a slash followed by the number of one-bits in the mask, or as a dotted quad.

The **no interface ethernet** *id* **ip address** command allows you to delete a particular subnet, all secondary subnets, or all subnets associated with the specified Ethernet interface.

Examples:

The following are equivalent ways to set the primary subnet of the Ethernet interface to 192.168.1.1 with a Class C subnet mask:

```
interface ethernet 0 ip address 192.168.1.1/24

interface ethernet 0 ip address 192.168.1.1 255.255.255.0
```

To set a secondary subnet of the Ethernet interface to 207.1.1.16/28 (with four host bits):

```
interface ethernet 0 ip address 207.1.1.16/28 secondary
```

To delete a particular subnet from the list of subnets, specify the particular subnet:

```
no interface ethernet 0 ip address 207.1.1.16/28
```

To delete all secondary subnets:

```
no interface ethernet 0 ip address secondary
```

To delete all subnets:

```
no interface ethernet 0 ip address
```

To show the IP subnets of the Ethernet interface:

```
show interface ethernet 0 ip address
```

interface ethernet *id* **ip dhcp client mode** { **standard** | **copper-mountain** | **cmn** }
show interface ethernet *id* **ip dhcp client mode**

These commands allow you to set or show the router's DHCP mode, whether **standard** or **copper-mountain**.

The connection profile, default profile, and IP configuration structures now include a **dhcp client mode** setting that selects between the **standard** RFC 2131 standards-based mode of operation (the default), and the **copper-mountain** or **cmn** proprietary mode of operation.

When the DHCP client is activated on a RFC1483 MER interface, it examines the **dhcp client mode** in the associated connection profile (or the default profile there was no explicitly configured connection profile). If the **dhcp client mode** specifies **standard**, the DHCP client initializes the htype and hlen fields in the header of its DHCP requests to the appropriate values for an RFC1483 MER interface (htype = 1 and hlen = 6). If the **dhcp client mode** specifies **copper-mountain** or **cmn**, the DHCP client initializes the htype and hlen fields in the header of its DHCP requests to zero.

When the DHCP client is activated on an Ethernet WAN interface, it examines the **dhcp client mode** in the associated IP configuration structure, and behaves as described above for the RFC1483 MER DHCP client.

Note: **cmn** is accepted as a synonym for **copper-mountain**.

---

interface ethernet *id* **ip multicast-fwd** { **yes** | **no** }
no interface ethernet *id* **ip multicast-fwd**
show interface ethernet *id* **ip multicast-fwd**

These commands allow you to set, disable, or show the multicast forwarding behavior on the specified Ethernet interface.

---

interface ethernet *id* **ip igmp-version** { **v1** | **v2** | **v3** }
show interface ethernet *id* **ip igmp-version**

These commands allow you to set or show the IGMP version to be used on the specified Ethernet interface. Beginning with Firmware version 8.7, **v3** is the default.

---

interface ethernet *id* **mac address** { *MAC-address* | **default** }
show interface ethernet *id* **mac address**

The first command allows you to set the MAC Address for the specified interface. You can return it to the default by typing in a MAC Address consisting of all zeros or by typing **default**. The **show** command applies to the LAN of *all* models, as well as the WAN on Ethernet WAN models.

---

interface ethernet *id* **mode** { **autonegotiate** | **100full** | **100half** | **10full** | **10half** |
100full-fixed | **100half-fixed** | **10full-fixed** | **10half-fixed** }
show interface ethernet *id* **mode**

Note: These commands are supported beginning with firmware version 8.2.

These commands allow you to set or show the Ethernet speed and duplex configuration to be used on the specified Ethernet interface. These commands only apply to 3300-Series products, single port Ethernet interface on either LAN or WAN. The default is auto-negotiation.

interface ethernet *id* **ip netbios proxy enable** { **yes** | **no** }
no interface ethernet *id* **ip netbios proxy enable**
show interface ethernet *id* **ip netbios proxy enable**

These commands allow you to enable, disable, or show the NetBIOS proxy status for the specified Ethernet interface. The NetBIOS proxy enables the ability to forward Windows Networking NetBIOS broadcasts. This is useful for, for example, a Virtual Private Network, in which you want to be able to browse the remote network to which you are tunnelling, as part of your Windows Network Neighborhood.

Routed connections, such as VPNs, can not use NetBEUI to carry the Network Neighborhood information. They need to use NetBIOS, because NetBEUI cannot be routed. This feature will allow browsing the Network Neighborhood without any additional workstation configuration.

Note: Microsoft Network browsing is available with or without a Windows Internet Name Service (WINS) server. Shared volumes on the remote network are accessible with or without a WINS server. Local LAN shared volumes that have Port Address Translation (PAT) applied to them are *not* available to hosts on the remote LAN. For tunnelled traffic, NAT on the WAN has no effect on the Microsoft Networking traffic.

interface ethernet *id* **address-serve dhcp enable** { **yes** | **no** }
no interface ethernet *id* **address-serve dhcp enable**
show interface ethernet *id* **address-serve dhcp enable**

These commands allow you to enable, disable, or show the DHCP IP address serving behavior of the specified Ethernet interface. These commands do not affect the DHCP server mode. Consequently, if the router is set to DHCP relay these commands have no effect.

The **show interface ethernet** *id* **address-serve dhcp** command may also include the following keywords: **available**, **leased**, **offered**, and **reserved**. These return the count of client IP addresses in their respective states.

Examples:

```
show interface ethernet 0 address-serve dhcp report available

show interface ethernet 0 address-serve dhcp report leased

show interface ethernet 0 address-serve dhcp report offered

show interface ethernet 0 address-serve dhcp report reserved
```

interface ethernet *id* **address-serve dhcp dns [1 | 2]** *ip-addr*

Note: This command is supported beginning with firmware version 8.5.

This command allows you to specify the IP addresses of primary and secondary DNS servers served to the client for this interface. If they are not specified, the globally configured (or derived) DNS addresses are served to the client instead.

These DNS addresses are not used internally by the router; the globally configured DNS addresses are used instead.

interface ethernet *id* **address-serve dhcp option 150 address** *www.xxx.yyy.zzz*
no interface ethernet *id* **address-serve dhcp option 150 address** *www.xxx.yyy.zzz*
show interface ethernet *id* **address-serve dhcp option 150 address**

Note: These commands are supported beginning with firmware version 8.6.

These commands allow you to configure, remove, or show up to four TFTP IP addresses per ALAN to be served via option 150.

interface ethernet *id* **ip rip exclude-wan-routes**
no interface ethernet *id* **ip rip exclude-wan-routes**
show interface ethernet *id* **ip rip exclude-wan-routes**

Note: These commands are supported beginning with firmware version 8.7.

These commands allow you to specify, disable, or show the status of broadcasting WAN routes via RIP. This is available only if **rip transmit** is enabled for the interface. The default is **no**, but if enabled, will drop any RIP routes with non-LANside information from RIP updates sent over the interface.

interface ethernet *id* **ip rip receive** { **no** | **v1** | **v2** | **both** | **v2-md5** }
no interface ethernet *id* **ip rip receive**
show interface ethernet *id* **ip rip receive**

These commands allow you to set, delete, or show the RIP receive behavior of the specified Ethernet interface.

Example:

```
show interface ethernet 0 ip rip receive
```

interface ethernet *id* **ip rip transmit** { **no** | **v1** | **v2broadcast** | **v2multicast** | **v2broadcast-md5** | **v2multicast-md5** }
no interface ethernet *id* **ip rip transmit**
show interface ethernet *id* **ip rip transmit**

These commands allow you set, delete, or show the RIP transmit behavior of the specified Ethernet interface.

Examples:

```
show interface ethernet 0 ip rip transmit
```

interface ethernet *id* **ip rip auth key** *id*
no interface ethernet *id* **ip rip auth key** *id*
show config interface ethernet *id* **ip rip auth key**

These commands allow you to create, delete, or show the RIP-2 Authentication key(s) on the specified interface.

---

interface ethernet *id* **ip rip auth key** *id* **start date** *date*
show interface ethernet *id* **ip rip auth key** *id* **start date**

These commands allow you to set or show a start date for the RIP-2 Authentication key(s) on the specified interface.

---

interface ethernet *id* **ip rip auth key** *id* **start time** *time*
show interface ethernet *id* **ip rip auth key** *id* **start time**

These commands allow you to set or show a start time for the RIP-2 Authentication key(s) on the specified interface.

---

interface ethernet *id* **ip rip auth key** *id* **end date** *date*
show interface ethernet *id* **ip rip auth key** *id* **end date**

These commands allow you to set or show an end date for the RIP-2 Authentication key(s) on the specified interface. The acceptable year range is from 1904 – 2039.

---

interface ethernet *id* **ip rip auth key** *id* **end time** *time*
show interface ethernet *id* **ip rip auth key** *id* **end time**

These commands allow you to set or show an end time for the RIP-2 Authentication key(s) on the specified interface.

---

interface ethernet *id* **rip auth key** *id* **end time mode** { **infinite** | **date** }
show interface ethernet *id* **ip rip auth key** *id* **end time mode**

These commands allow you to set or show the end time mode for the RIP-2 Authentication key(s) on the specified interface. **date** specifies that an expiration date and time will be used; **infinite** specifies that the key will never expire.

---

interface ethernet *id* **rip auth key** *id* **key** <*string*>

These commands allow you to assign a RIP-2 Authentication key. Keys must be manually entered and must consist of 1 – 16 ASCII characters each.

---

interface ethernet *id* **pppoe enable** { **yes** | **no** }
no interface ethernet *id* **pppoe enable**
show interface ethernet *id* **pppoe enable**

These commands allow you enable, disable, or show the PPP over Ethernet behavior of the specified interface.

---

show interface ethernet *id* **statistics**
show interface ethernet *id* **stats**

These commands allow you to display statistics for the specified Ethernet interface, including receive frames, octets, and errors, and transmit frames, octets, and errors.

interface ethernet *id* **ip filterset** *fs-id*
no interface ethernet *id* **ip filterset**
show interface ethernet *id* **ip filterset**

These commands allow you to enable, disable, or show an IP filterset identified by *fs-id* on the specified Ethernet interface. *fs-id* is specified as an ASCII string corresponding to the name of a filterset. See "IP Filterset Configuration Commands" on page 2-121 for more information.

interface ethernet *id* **ip nat enable** { **yes** | **no** }
no interface ethernet *id* **ip nat enable**
show interface ethernet *id* **ip nat enable**

These commands allow you to enable, disable, or show the Network Address Translation behavior for the specified WAN interface.

interface ethernet *id* **ip nat map-list** *list-tag*
no interface ethernet *id* **ip nat map-list**
show interface ethernet *id* **ip nat map-list**

These commands allow you to set, delete, or show a NAT map list for the specified WAN interface.

interface ethernet *wan-id* **ip nat passthrough enable** { **yes** | **no** }
no interface ethernet *id* **ip nat passthrough enable**
show interface ethernet *id* **ip nat passthrough enable**

Note: These commands are supported beginning with firmware version 8.2.

These commands allow you to enable, disable, or show the NAT passthrough behavior for the specified WAN interface. The IP passthrough feature allows for a single LAN PC to have the router's public address assigned to it, in addition to providing PAT (NAPT) via the same public IP address for all other hosts on the private LAN subnet.

interface ethernet *wan-id* **ip nat passthrough dhcp enable** { **yes** | **no** }
no interface ethernet *wan-id* **ip nat passthrough dhcp enable**
show interface ethernet *wan-id* **ip nat passthrough dhcp enable**

Note: These commands are supported beginning with firmware version 8.2.

These commands allow you to enable, disable, or show the NAT passthrough DHCP behavior for the specified WAN interface. This governs DHCP addressing for the passthrough host.

interface ethernet *wan-id* **ip nat passthrough dhcp mac-address** { *mac-address* }
show interface ethernet *wan-id* **ip nat passthrough dhcp mac-address**

Note: These commands are supported beginning with firmware version 8.2.

These commands allow you to set or show the NAT passthrough DHCP MAC address for the specified WAN interface. This specifies the MAC address of the passthrough host.

interface ethernet *id* **ip nat server-list** *list-tag*
no interface ethernet *id* **ip nat server-list**
show interface ethernet *id* **ip nat server-list**

These commands allow you to set, delete, or show a NAT server list for the specified WAN interface.

interface ethernet *wan-id* **mac address** { *MAC-address* | **default** }
show interface ethernet *wan-id* **mac address**

The first command allows you to set the MAC Address for the WAN on a WAN Ethernet Router. You can return it to the default by typing in a MAC Address consisting of all zeros or by typing **default**. The **show** command applies to the LAN of *all* models, as well as the WAN on the R9100 and R910.

interface ethernet *id* **address-serve clients** { **any** | **none** | { **bootp** | **dhcp** | **macip** | **wan** } }
no interface ethernet 0 address-serve clients { **any** | { **bootp** | **dhcp** | **macip** | **wan** }$^{+}$ }
show interface ethernet 0 address-serve clients

The **interface ethernet** *id* **address-serve clients** command allows you to configure the types of clients that may request IP addresses from the address server. If you specify the keyword **any**, the address server will accept requests from clients of any type supported by the router. Otherwise, you may specify one or more of the keywords **bootp**, **dhcp**, **macip**, or **wan**, in which case the address server will accept requests from only the specified types of clients. If you specify the keyword **none**, the address server will not accept requests from clients of any type.

The **no interface ethernet** *id* **address-serve clients** command removes the specified client types from those from which the address server will accept requests.

interface ethernet *id* **address-serve dhcp lease-time** *hours*
show interface ethernet *id* **address-serve dhcp lease-time**

These commands allow you to set or show the address serving DHCP lease time to any number of hours, up to and including 168 (one week). The default DHCP lease time is one hour.

show interface ethernet *id* **address-serve dhcp addresses**

Note: This command is supported beginning with firmware version 8.5.

This command allows you to display the ethernet IP addresses being served via DHCP, and the host name of the served device, if available.

interface ethernet *id* **ip dhcp client** [**renew** | **release**]

Note: This command is supported beginning with firmware version 8.5.

This command allows you to renew or release the ethernet WAN IP address lease being served via DHCP.

interface ethernet *id* **address-serve dhcp next-server** *ip-addr*

Note: This command is supported beginning with firmware version 8.5.

This command specifies the IP address of the next server in the boot process, typically a Trivial File Transfer Protocol (TFTP) server.

---

show interface ethernet *id* **ip dhcp client status**

Note: This command is supported beginning with firmware version 8.5.

This command allows you to display the status of the ethernet WAN being served via DHCP. It displays:

IP Address	IP Subnet Mask
IP Gateway	DHCP server
DNS server 1	DNS server 2 (if any)
Lease Expiration	

---

interface ethernet *id* **address-serve gateway** { *gw-ip-addr* | **default** { *ip-addr/mask-bits* | *ip-addr mask* } }
show interface ethernet *id* **address-serve gateway**

This command allows you to specify the gateway IP address that will be served to clients requesting an address via an address serving protocol that can serve a gateway address. You may specify a gateway IP address for each Ethernet subnet for which you have configured an address-serving pool. (See the description of the **interface ethernet** *id* **address-serve range** command on page 2-32.)

If you specify the keyword **default**, you must also specify an Ethernet subnet; the gateway IP address for the specified subnet will be reset to its default value. The default gateway IP address for a particular subnet is either the router's default gateway (if that gateway is on the specified subnet) or the router's address on the subnet.

---

interface ethernet *id* **address-serve helper** *ip-addr*
no interface ethernet *id* **address-serve helper** [*ip-addr*]
show interface ethernet *id* **address-serve helper**

These commands allow you to configure or display the addresses of up to four remote DHCP servers to which the router will forward DHCP requests when it is acting as a DHCP relay agent. The **interface ethernet** *id* **address-serve helper** command adds the specified IP address to the server list. The **no interface ethernet** *id* **address-serve helper** command removes the specified IP address from the server list; if you omit the IP address, *all* configured DHCP server IP addresses are removed.

Examples:

```
#show interface ethernet 0 address-serve helper
#interface ethernet 0 address-serve helper 10.0.0.1
#interface ethernet 0 address-serve helper 20.0.0.1
#interface ethernet 0 address-serve helper 30.0.0.1
#no interface ethernet 0 address-serve helper 20.0.0.1
#show interface ethernet 0 address-serve helper
interface ethernet 0 address-serve helper 10.0.0.1
```

```
interface ethernet 0 address-serve helper 30.0.0.1
#
```

---

interface ethernet *id* **address-serve mode** { **relay** | **server** }
show interface ethernet *id* **address-serve mode**

These commands allow you to specify or display the address serving mode for the specified Ethernet interface. The keyword **relay** causes the router to act as a DHCP relay agent. The keyword **server** enables address serving from one or more locally configured address pools.

Examples:

```
#interface ethernet 0 address-serve mode server
#show interface ethernet 0 address-serve mode
interface ethernet 0 address-serve mode server
#
```

---

interface ethernet *id* **address-serve range** { **auto** | *from-addr to-addr* }
no interface ethernet *id* **address-serve range** *from-addr to-addr*
show interface ethernet *id* **address-serve range**

This command configures a pool of IP addresses for use by the address server. You may specify one address pool for each configured Ethernet subnet (primary and secondary). The total number of addresses in all configured pools may not exceed 512 addresses.

If you specify the keyword **auto** instead of an IP address range, the router will automatically configure IP address pools for each configured Ethernet subnet. An automatically configured pool will include one-half of the number of addresses available in the corresponding subnet, and will be located in the opposite half of the subnet from the router's IP address on the subnet. If the total number of addresses required would exceed the maximum of 512 total addresses, the 512 available addresses will be allocated on a pro-rata basis across all pools.

---

interface ethernet *id* **address-serve** { **no** | **off** | **on** | **yes** }
no interface ethernet *id* **address-serve**
show interface ethernet *id* **address-serve**

These commands enable, disable, or display the status of address-serving for the specified Ethernet interface.

Stateful Inspection Configuration Commands

See also:

- "Stateful Inspection Commands" on page 3-21 for Connection Profile commands.
- "Stateful Inspection Commands" on page 2-85 for Global Stateful Inspection commands.

Note: The commands in this section are supported beginning with firmware version 8.2.

---

interface ethernet *id* **ip state-insp enable** { **yes** | **no** | **on** | **off** }
no interface ethernet *id* **ip state-insp**
show interface ethernet *id* **ip state-insp enable**

These commands allow you to set, disable, or show the status of stateful inspection for the specified interface. This option is disabled by default. Stateful inspection prevents unsolicited inbound access when NAT is disabled.

---

interface ethernet *id* **ip state-insp router-access** { **yes** | **no** | **on** | **off** }
no interface ethernet *id* **ip state-insp router-access**
show interface ethernet *id* **ip state-insp router-access**

These commands allow you to set, disable, or show the status of default mapping to router for the specified interface.

---

interface ethernet *id* **ip state-insp tcp-seq-diff** *diff*
show interface ethernet *id* **ip state-insp tcp-seq-diff**

These commands allow you to set or show TCP sequence difference acceptable for the specified interface. The TCP sequence number difference maximum allowed value is 65535. If the value of **tcp-seq-diff** is 0, it means that this check is disabled.

---

interface ethernet *id* **ip state-insp deny-frag** { **yes** | **no** | **on** | **off** }
no interface ethernet *id* **ip state-insp deny-frag**
show interface ethernet *id* **ip state-insp deny-frag**

These commands allow you to set, disable, or show whether fragmented packets are received for the specified interface.

---

interface ethernet *id* **ip state-insp xposed-list** *xposed-list_name*
no interface ethernet *id* **ip state-insp xposed-list**
show interface ethernet *id* **ip state-insp xposed-list**

These commands allow you to set, disable, or show the status of a stateful inspection exposed address list for the specified interface. Exposed address lists are similar to NAT server lists. Exposed addresses in the list will not be subject to stateful inspection and hence unsolicited inbound traffic will be allowed to these addresses.

These are active only if NAT is disabled on the profile.

Ethernet Interface Static Client Address Translation Commands

Note: The commands in this section are supported beginning with firmware version 8.5.

---

interface ethernet *lan_interface_id* **scat enable** [**yes** | **no**]
show interface ethernet *lan_interface_id* **scat enable**

These commands allow you to enable, disable, or show the status of static client address translation on the specified LAN interface. This feature allows a statically addressed host whose address falls outside of the LAN subnet(s) to simply plug in and get online without any manual configuration on either the host or the Motorola Netopia® Router.

If **scat enable** is set to **yes**, statically addressed LAN hosts that have an address outside of LAN subnets will be able to communicate via the Router's WAN interface to the Internet.

Supported static IP address values *must* fall *outside* of the Router's LAN subnet(s).

Ethernet Interface VRRP Commands

Note: The commands in this section are supported beginning with firmware version 8.5.

A Virtual Router is a software abstraction consisting of a group of two or more hardware routers protecting one or more IP addresses. One of the routers is designated as the *Master*, while the others are backups. VRRP is a protocol that provides redundancy to routers within a local area network by allowing alternate paths for a PC without changing the IP address or MAC address by which the PC knows its gateway.

A Virtual Router cannot be enabled unless it is populated with a Virtual ID and a Virtual IP address. The Virtual Router index starts from 1 (one). Two virtual routers can be configured per LAN interface.

---

interface ethernet *id* **ip vrrp vrouter** *id* **vrid** [**1... 255**]
show interface ethernet *id* **ip vrrp vrouter** *id* **vrid**

These commands allow you to specify or display an ID (**vrid**) for the Virtual Router. **vrid** values may be in the range 1 – 255.

---

interface ethernet *id* **ip vrrp vrouter** *id* **vrip** *ip-addr*
show interface ethernet *id* **ip vrrp vrouter** *id* **vrip**

These commands allow you to specify or display an IP address for the Virtual Router.

---

interface ethernet *id* **ip vrrp vrouter** *id* **priority** [**1... 255**]
show interface ethernet *id* **ip vrrp vrouter** *id* **priority**

These commands allow you to specify or display a priority for the Virtual Router. The default value is 100, if not the owner of the virtual IP address. **priority** values may be in the range 1 – 255.

---

interface ethernet *id* **ip vrrp vrouter** *id* **adv-intvl** [**1... 255**]
show interface ethernet *id* **ip vrrp vrouter** *id* **adv-intvl**

These commands allow you to specify or display an advertisement interval in seconds. The default value is one second; **adv-intvl** values may be in the range 1 – 255.

---

interface ethernet *id* **ip vrrp vrouter** *id* **preempt-mode enable** [**no** | **yes** | **on** | **off**]
show interface ethernet *id* **ip vrrp vrouter** *id* **preempt-mode enable**

These commands allow you to enable, disable, or display the status of preempt mode. The default is enabled.

---

interface ethernet *id* **ip vrrp vrouter** *id* **enable** [**no** | **yes** | **on** | **off**]
show interface ethernet *id* **ip vrrp vrouter** *id* **enable**

These commands allow you to enable, disable, or display the status of the Virtual Router.

---

no interface ethernet *id* **ip vrrp vrouter** *id*

This command allows you to delete a Virtual Router.

---

show interface ethernet *id* **ip vrrp wan-monitor enable** [{ **yes** | **no** | **on** | **off** }]
show interface ethernet *id* **ip vrrp wan-monitor enable**

These commands allow you to enable, disable, or display the status of the WAN monitor.

---

interface ethernet *id* **ip vrrp master-dhcp-srv enable** [{ **yes** | **no** | **on** | **off** }]
show interface ethernet *id* **ip vrrp master-dhcp-srv enable**

These commands allow you to enable, disable, or display the status of the DHCP behavior. The Virtual Router can either serve or relay DHCP only if it is in Master state.

---

interface ethernet *id* **ip vrrp dhcp-gateway** *ip-addr*
show interface ethernet *id* **ip vrrp dhcp-gateway**

These commands allow you to specify or display the Virtual Router DHCP gateway IP address.

Virtual LAN (VLAN) configuration commands

Note: See also "RADIUS Authentication Profile configuration commands" on page 2-39 and "Additional LAN configuration command" on page 2-16.

VLAN Configuration Commands
vlan *id* **by** [**port-based** | **global**] **no vlan** *id* **vlan** *id* **name** *name* **vlan** *id* **network** { **none** | **lan** | **eth2** | **eth3** | **eth4** | **eth5** | **eth6** | **eth7** } **vlan** *id* **id** { **1** .. **4094** } (supported in V8.6.1) **vlan** *id* **8021x authprofile** { **authprofile tag** *name* | **authprofile** *id* } **no vlan** *id* **8021x authprofile** **vlan** *id* **interface eth** { **1** | **2** | **0/1** | **0/...***n* } **tag** { **yes** | **no** } [**tos-priority** { **off** | **on** -}] [**iptos-promote** { **off** | **on** }] [**authprofile** { **name** | **id** }] [**inter-vlan-routing** { **group-1**... **group-8** } **enable** { **yes** | **no** }] **no vlan** *id* **interface eth** { **1** | **2** | **0/1** | **0/...***n* } **vlan** *id* **interface ssid** *n* **tag** { **yes** | **no** } **no vlan** *id* **interface ssid** *n* **vlan** *id* **interface usb 0 tag** { **yes** | **no** } **no vlan** *id* **interface usb 0** **vlan** *id* **interface cp** *n* **tag** { **yes** | **no** } **no vlan** *id* **interface cp** *n* **show config vlan** { *id* }

vlan *id* **by** [**port-based** \| **global**]
no vlan *id*

These commands allow you to create or delete a VLAN specified by *id* and designate it either **port-based** or **global**. You can create up to 16 VLANs.

---

vlan *id* **name** *name*

This command allows you to assign a free-form name *name* to a VLAN specified by *id*.

---

vlan *id* **network** { **none** | **lan** | **eth2** | **eth3** | **eth4** | **eth5** | **eth6** | **eth7** }

This command allows you to define what additional LAN (ALAN) network is associated with the VLAN specified by *id*.

---

vlan *id* **id** { **1** .. **4094** }

This command allows you to change the VLAN ID, which will effectively require you to refer to the VLAN by its new VID after issuing this command.

Beginning with Firmware Version 8.7, a VID of zero (0) is permitted on the Ethernet WAN port only.

---

vlan *id* **8021x authprofile** { **authprofile tag** *name* | **authprofile** *id* }

This command allows you to enable 802.1x authentication for the VLAN specified by *id*. This option is only supported on Router models with VGx technology or single Ethernet port models. If you are configuring a VLAN for a Motorola Netopia® Router model with VGx technology (wired or wireless), you can specify a RADIUS server authentication profile for user authentication. This command allows you to associate a VLAN with an 802.1x RADIUS authentication profile. You must create an authentication profile, if you have not already done so. See "RADIUS Authentication Profile configuration commands" on page 2-39.

Note: If you enable 802.1x for a VLAN that includes a wireless SSID, you *must* set **wireless privacy** to **WPA-802.1x** as well. See "Wireless Privacy Commands (new and revised)" on page 2-91. If multiple SSIDs are split across several VLANs, the VLANs must either:

- *all* have 802.1x enabled with WPA-802.1x enabled in Wireless Privacy, or
- have the VLANs set to 802.1x *dis*abled and **wireless privacy** set to some other privacy setting. In that case **wireless privacy** can be any setting.

Wireless does not currently support separate privacy modes per SSID. When enabling WPA-802.1x, wireless will default to the RADIUS configuration (see "RADIUS Authentication Profile configuration commands" on page 2-39), unless it is part of a VLAN. If it is part of a VLAN it will use the VLAN authentication profile's specified RADIUS server.

---

vlan *id* **interface eth** { **1** | **2** | **0/1** | **0/...***n* } **tag** { **yes** | **no** }
[**tos-priority** { **off** | **on** -}]
[**iptos-promote** { **off** | **on** }]
[**authprofile** { **name** | **id** }]
[**inter-vlan-routing** { **group-1**... **group-8** } **enable** { **yes** | **no** }]
no vlan *id* **interface eth** { **1** | **2** | **0/1** | **0/...***n* }

These commands allow you to create or delete a VLAN specified by *id* on an Ethernet interface indicated by its interface number. The option **eth 1** = Ethernet LAN; **eth 2** = Ethernet WAN, where applicable. If the Motorola Netopia® Router model is a non-VGx model, the only available Ethernet port is numbered **0/1**; for multiple managed-switch VGx models the number *n* is the number of the physical Ethernet port.

- **tos-priority** - allows you to enable or disable packet prioritization based on any 802.1p priority bits in the VLAN header to prioritize packets within the Router's internal queues, according to DiffServ priority

mapping rules.

- **iptos-promote** - allows you to enable or disable the translation of 802.1p priority bits to and from the IP-TOS header bit field. When enabled, write any 802.1p priority bits into the IP-TOS header bit field for *received* IP packets on this port destined for this VLAN; and write any IP-TOS priority bits into the 802.1p priority bit field for tagged IP packets *transmitted* from this port for this VLAN. All mappings between Ethernet 802.1p and IP-TOS are made according to a pre-defined QoS mapping policy.
- **authprofile** - allows you to associate this VLAN with an 802.1x authentication profile specified by *name* or *id*.
- **inter-vlan-routing** - (supported beginning with Firmware Version 8.7.4) when set to **yes**, allows you to associate this VLAN with an inter-VLAN routing group such that the specified VLAN can communicate with another VLAN in the same group. VLANS that are not associated with the same inter-VLAN routing group cannot communicate with each other.

---

vlan *id* **interface ssid** *n* **tag** { **yes** | **no** }
no vlan *id* **interface ssid** *n*

These commands allow you to create or delete a port-based VLAN specified by *id* on a wireless SSID, if available, indicated by its SSID number.

---

vlan *id* **interface usb 0 tag** { **yes** | **no** }
no vlan *id* **interface usb 0**

These commands allow you to create or delete a port-based VLAN specified by *id* on the Router's USB port, if available.

---

vlan *id* **interface cp** *n* **tag** { **yes** | **no** }
no vlan *id* **interface cp** *n*

These commands allow you to create or delete a port-based VLAN specified by *id* on the Router's console port, if available, indicated by its port number *n*.

---

show config vlan { *id* }

This command allows you to display the configuration of all VLANs or a particular VLAN specified by *id*.

RADIUS Authentication Profile configuration commands

Note: The commands in this section are supported beginning with Firmware Version 8.4.2.

Authentication Profile Configuration Commands
authprofile *id* [**yes** | **no**]
authprofile *id* **tag** *string*
authprofile *id* **remote server** *string*
authprofile *id* **remote secret** *string*
authprofile *id* **alternate server** *string*
authprofile *id* **alternate secret** *string*
authprofile *id* **radius identifier** *string*
authprofile *id* **radius port** { **1** ...**65535** }
show config authprofile *id*

authprofile *id* [**yes** \| **no**]

This command allows you to create or delete an authentication profile identified by *id* containing relevant information to access a RADIUS server. You associate the profile with a VLAN using the **vlan 8021x authprofile** command (page 37).

authprofile *id* **tag** *string*

This command allows you to name an authentication profile identified by *id* with a free-form name of up to 32 characters.

authprofile *id* **remote server** *string*

This command allows you to specify the RADIUS server's IP address or fully qualified server name.

authprofile *id* **remote secret** *string*

This command allows you to specify the RADIUS server CHAP secret.

authprofile *id* **alternate server** *string*

This command allows you to specify an alternate RADIUS server, if available.

authprofile *id* **alternate secret** *string*

This command allows you to specify the alternate RADIUS server CHAP secret.

authprofile *id* **radius identifier** *string*

This command allows you to specify the RADIUS Network Access Server (NAS) identifier. The default NAS identifier is an ASCII representation of the server's base MAC address.

authprofile *id* **radius port** { **1** ...**65535** }

This command allows you to specify the RADIUS server's port number. Ordinarily, the RADIUS server port number is 1812. If you are using a different port number, enter it here.

show config authprofile *id*

This command allows you to display the configuration of an authentication profile identified by *id*.

NetBIOS configuration commands

NetBIOS Configuration Commands
interface ethernet 0 address-serve netbios mode type { **b-node** | **p-node** | **m-node** | **h-node** } **show interface ethernet 0 address-serve netbios mode type** **interface ethernet 0 address-serve netbios mode enable** { **yes** | **no** } **no interface ethernet 0 address-serve netbios mode enable** **show interface ethernet 0 address-serve netbios mode enable** **interface ethernet 0 address-serve netbios scope enable** { **yes** | **no** } **no interface ethernet 0 address-serve netbios scope enable** **show interface ethernet 0 address-serve netbios scope enable** **interface ethernet 0 address-serve netbios scope name** *domain-name* **show interface ethernet 0 address-serve netbios scope name** **interface ethernet 0 address-serve netbios name-server enable** { **yes** | **no** } **no interface ethernet 0 address-serve netbios name-server enable** **show interface ethernet 0 address-serve netbios name-server enable** **interface ethernet 0 address-serve netbios name-server address** *xxx.xxx.xxx.xxx* [**secondary**] **show interface ethernet 0 address-serve netbios name-server address**

interface ethernet 0 address-serve netbios mode enable { **yes** \| **no** }
no interface ethernet 0 address-serve netbios mode enable
show interface ethernet 0 address-serve netbios mode enable

These commands allow you to enable, delete, or show the router's IP address serving capability on the Ethernet interface in NetBIOS mode.

interface ethernet 0 address-serve netbios mode type { **b-node** \| **p-node** \| **m-node** \| **h-node** }
show interface ethernet 0 address-serve netbios mode type

These commands allow you to set or show the router's NetBIOS mode type of IP address serving on the Ethernet interface.

interface ethernet 0 address-serve netbios scope enable { **yes** \| **no** }
no interface ethernet 0 address-serve netbios scope enable
show interface ethernet 0 address-serve netbios scope enable

These commands allow you to set, delete, or show whether NetBIOS scope is enabled.

---

interface ethernet 0 address-serve netbios scope name *domain-name*
show interface ethernet 0 address-serve netbios scope name

These commands allow you to set or show the domain name under which the NetBIOS scope is enabled.

---

interface ethernet 0 address-serve netbios name-server enable { **yes** | **no** }
no interface ethernet 0 address-serve netbios name-server enable
show interface ethernet 0 address-serve netbios name-server enable

These commands allow you to set, delete, or show whether a NetBIOS name server address is served to NetBIOS clients.

---

interface ethernet 0 address-serve netbios name-server address *xxx.xxx.xxx.xxx* [**secondary**]
show interface ethernet 0 address-serve netbios name-server address

These commands allow you to set or show the IP address of the NetBIOS name server.

If the keyword **secondary** is specified and there is no primary WINS server the command will be rejected as **CLI_NO_CFG_SUPPORT_ERR**, with the error message "; error 2: not supported with current configuration“.

Generic WAN Interface configuration commands

Note: For possible values of *intf-type*, refer to "Interface Naming Conventions" on page 1-3. Generic WAN Interface Commands may be applied to any router WAN interface by specifying the *intf-type* **wan** together with the appropriate interface *id*. Alternatively, you can specify the more specific *intf-type* if you choose.

Generic WAN Interface Configuration Commands
interface *intf-type id* **dle** { **hdlc** | **ppp** [{**vcmux** | **vcmultiplexed** | **llcsnap**}] | **rfc1483** [{ **bridged** | **routed** }] | **rfc1490** } **show interface** *intf-type id* **dle** **show interface** *intf-type id* **statistics** **show interface** *intf-type id* **stats** **show interface wan** *id* **status** **interface wan 0 tracking** { **yes** | **no** }
Restricted WAN Interface Configuration Commands
interface { **adsl** | **ethernet** | **isdn** | **sdsl** } *id* **pppoe enable** { **yes** | **no** } **no interface** { **adsl** | **ethernet** | **isdn** | **sdsl** } *id* **pppoe enable** **show interface** { **adsl** | **ethernet** | **isdn** | **sdsl** } *id* **pppoe enable** **interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } { **yes** | **no** } **interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **pcr** *num* **show interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **pcr**

interface *intf-type id* **dle** { **hdlc** \| **ppp** [{**vcmux** \| **vcmultiplexed** \| **llcsnap**}] \|
rfc1483 [{ **bridged** \| **routed** }] \| **rfc1490** }
show interface *intf-type id* **dle**

These commands allow you to set or show the global data link encapsulation type of the interface specified by *intf-type id*. At this time you can generally think of the data link encapsulation of interface 1 as the global data link encapsulation of the router itself.

Note: **atmfuni** is accepted as a synonym for **rfc1483** and **frame-relay** is accepted as a synonym for **rfc1490**. For **ppp**, the default mode is **vcmux**. For **rfc1483**, the default mode for frame-based SDSL (R7100) interfaces is **bridged**, while the default mode for cell-based SDSL (R7200) interfaces is **routed**.

Example:

```
interface wan 1 dle frame-relay
```

show interface *intf-type id* **statistics**
show interface *intf-type id* **stats**

These commands allow you to display statistics for the specified interface, including receive frames, octets, and errors, and transmit frames, octets, and errors. For switched ISDN interfaces, the statistics are broken down by channel.

show interface wan *id* **status**

Note: This command is supported beginning with firmware version 8.5.

This command allows you to view the general status of the WAN. It displays:

- Interface type
- MAC address
- IP Address
- Status (Down, Activating, or Connected)

interface wan 0 tracking { **yes** | **no** }

For D-Series CSU/DSU equipment, this command allows you to track or not track the primary interface speed. Specifying **yes** means the primary interface (AUX) speed will be tracked, which is the default. Changing this to **no** currently means that we will be running at 1.5 MHz.

Restricted WAN Interface configuration commands

interface { **adsl** | **ethernet** | **isdn** | **sdsl** } *id* **pppoe enable** { **yes** | **no** }
no interface { **adsl** | **ethernet** | **isdn** | **sdsl** } *id* **pppoe enable**
show interface { **adsl** | **ethernet** | **isdn** | **sdsl** } *id* **pppoe enable**

These commands allow you enable, disable, or show the PPP over Ethernet behavior of the specified interface.

interface { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } { **yes** | **no** }

This command allows you to enable or disable a Permanent Virtual Circuit (PVC) on the specified ADSL or SDSL interface.

interface { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **pcr** *num*
show interface { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **pcr**

These commands allow you to assign or show a Peak Cell Rate value (PCR) on a specified Permanent Virtual Circuit (PVC) on an ADSL or SDSL interface.

Motorola Netopia® routers support two ATM classes of service for data connections: Unspecified Bit Rate (UBR) and Constant Bit Rate (CBR). You can configure these classes of service on a per VC basis. The default ATM class of service is UBR for data. The ATM class of service is not configurable for voice virtual circuits.

UBR VC: No configuration is needed for UBR VCs.
CBR VC: One parameter is required for CBR VCs, the Peak Cell Rate **pcr** that applies to the VC. This value should be between 1 and the line rate. You set this value according to specifications defined by your service provider.

ISDN WAN Interface configuration commands

ISDN WAN Interface Configuration Commands
Generic ISDN
interface isdn *id* **line type** { **switched** | **leased** | **idsl-ascend** | **idsl-cmn** } **show interface isdn** *id* **line type** **show interface isdn** *id* **status** [**b1** | **b2**]
Permanent ISDN (IDSL) only
interface isdn *id* **imux mode** { **mlppp** | **dml** } **show interface isdn** *id* **imux mode** **interface isdn** *id* **speed** { **b1** | **b2** | **2b** | **2b+d** } **show interface isdn** *id* **speed**
Switched ISDN only
interface isdn *id* **switch** { **auto** | **ni1** | **5esspttopt** | **5essmultipt** | **dms100** | **ts013** | **euroisdn** | **japanntt** | **uk-euro** } **interface isdn** *id* **dn** { **1** | **2** } *string* **no interface isdn** *id* **dn** { **1** | **2** } **show interface isdn** *id* **dn** { **1** | **2** } **interface isdn** *id* **spid** { **1** | **2** } *string* **no interface isdn** *id* **spid** { **1** | **2** } **show interface isdn** *id* **spid** { **1** | **2** }

---

interface isdn *id* **imux mode** { **mlppp** \| **dml** }
show interface isdn *id* **imux mode**

These commands allow you set or show the ISDN interface IMUX bonding mode: Multilink PPP or DML (for Copper Mountain Networks central office equipment).

Example:

```
interface isdn 1 imux mode dml
```

interface isdn *id* **line type** { **switched** | **leased** | **idsl-ascend** | **idsl-cmn** }
show interface isdn *id* **line type**

These commands allow you set or show the ISDN interface mode: switched, leased, idsl-ascend (IDSL for Lucent/Ascend Communications central office equipment), or idsl-cmn (IDSL for Copper Mountain Networks central office equipment).

Example:

```
interface isdn 1 mode leased
```

show interface isdn *id* **status** [**b1** | **b2**]

This command allows you display the status of the specified ISDN/IDSL interface. For a switched ISDN interface, you may specify the optional keyword **b1** or **b2**, in which case the status of the specified B-channel is displayed rather than the status of the interface itself.

For a leased ISDN/IDSL interface, the possible status strings and their meanings are:

Status String	Meaning
Inactive	The interface is not yet active.
Waiting for rate negotiation	The interface is in the process of sensing the data rate configured for the IDSL line at the central office. This status applies only to an interface set to idsl-cmn mode, in which the router can sense the data rate automatically.
Backup recovery in progress	The interface is in the process of recovering back to the primary interface from a backup interface after a failure.
Connected at xxx Kbps	The interface is connected to the DSLAM or other end device at the specified data rate. (xxx will be one of 64, 128, or 144.)

For a switched ISDN interface, the possible status strings and their meanings are:

Status String	Meaning
Inactive	The interface is not yet active.
Active	The interface is active, and this is an interface that does not require SPIDs.
Active, *n* of *m* SPIDs registered	The interface is active, and this is an interface that requires SPIDs. *n* indicates the number of SPIDs that have been successfully registered so far, and *m* indicates the total number of SPIDs to be registered. If *n* is less than *m*, the device is still in the process of registering some of the SPIDs.

Status String	Meaning
Active, *n* of *m* SPIDs registered (*p* failed)	The interface is active, and this is an interface that requires SPIDs. *n* indicates the number of SPIDs that have been successfully registered so far, *m* indicates the total number of SPIDs to be registered, and *p* indicates the number of SPIDs that failed registration. If the sum of *n* and *p* is less than *m*, the device is still in the process of registering some of the SPIDs.

For one of the B-channels of a switched ISDN interface, the possible status strings and their meanings are:

Status String	Meaning
Inactive	The associated interface is not yet active.
Idle	The channel is not currently in use.
Speech Call	The channel is in use by a speech call.
64 Kbps Data Call	The channel is in use for a 64 Kbps data call.
56 Kbps Data Call	The channel is in use for a 56 Kbps data call.
3.1 Khz Call	The channel is in use for a 3.1 Khz call.

Example:

```
#show interface isdn 1 status
Connected at 144 Kbps
```

---

interface isdn *id* **speed** { **b1** | **b2** | **2b** | **2b+d** }
show interface isdn *id* **speed**

These commands, which apply only to permanent ISDN (i.e., IDSL), allow you to set or show the data rate (and B Channel usage) of the ISDN line. **b1** means use Channel B1 at 64 Kbps, **b2** means use Channel B2 at 64 Kbps, **2b** means use both Channels B1 and B2 at 128 Kbps, and **2b+d** means use all three channels at 144 Kbps.

---

interface isdn *id* **switch** { **auto** | **ni1** | **5esspttopt** | **5essmultipt** | **dms100** | **ts013** | **euroisdn** | **japanntt** | **uk-euro** }

This command allows you to change the ISDN switch type. This command applies only to switched ISDN. The only currently supported *id* is 1, which identifies the ISDN interface in the WAN 1 slot of the Motorola Netopia® router. The WAN 2 slot (id 2) cannot be populated with an ISDN wanlet at this time, and the Motherboard (id 0) is incapable of supporting ISDN internally.

Under many circumstances it is unnecessary to explicitly set the switch type, particularly in Europe. This is because for “S/T” ISDN routers the default switch type is **euroisdn**, and for “U” ISDN routers the default switch type is **ni1**.

auto is appropriate only in the United States and allows the router to auto-determine the switch type, SPIDs, and directory numbers (DNs).

uk-euro sets the switch type to Euro-ISDN, and as a side effect sets the console's clock time display type to 24 hour (i.e., "17:45" instead of "5:45 PM").

Example:

The command to set the switch type of the wanlet for the correct value in Japan is:

```
interface isdn 1 switch japanntt
```

---

interface isdn *id* **dn** { **1** | **2** } *string*
no interface isdn *id* **dn** { **1** | **2** }
show interface isdn *id* **dn** { **1** | **2** }

These commands allow you to set, change, delete, or show the directory numbers associated with the specified ISDN interface. These commands apply only to switched ISDN. The only currently supported *id* is 1. The string parameter can contain up to 32 characters. Non-dialable characters are allowed (and are ignored).

Example:

```
interface isdn 1 dn 1234567
```

---

interface isdn *id* **spid** { **1** | **2** } *string*
no interface isdn *id* **spid** { **1** | **2** }
show interface isdn *id* **spid** { **1** | **2** }

These commands allow you to set, change, delete, or show the SPIDs associated with the specified ISDN interface. These commands apply only to switched ISDN. The only currently supported *id* is 1. The string parameter can contain up to 23 characters. Illegal characters are allowed (for instance, for formatting) and are ignored by the interface.

ADSL WAN Interface configuration commands

ADSL WAN Interface Configuration Commands
interface adsl *id* **pvc** *vpi-value vci-value* **show interface adsl** *id* **pvc** **show interface adsl** *id* **status** **show interface adsl** *id* **statistics** **interface adsl** *id* **signaling-mode** { **fdm** | **echo-cancellation** } **show interface adsl** *id* **signaling-mode** **interface adsl** *id* **trellis-coding** { **yes** | **no** } **show interface adsl** *id* **trellis-coding** **no interface adsl** *id* **trellis-coding**

interface adsl *id* **pvc** *vpi-value vci-value*
show interface adsl *id* **pvc**

These commands allow you to set, change, or show the PVC VPI and VCI values associated with the ADSL WAN interface.

show interface adsl *id* **status**

This command allows you to display the status of the specified ADSL interface. For an ADSL interface, the possible status strings and their meanings are:

Status String	Meaning
Connected at *xxx rx / yyy tx* Kbps	The interface is connected to the DSLAM at the specified speeds, where *xxx* is the downstream (receive) speed and *yyy* is the upstream (transmit) speed, each in Kbps.
Activation Backoff	The ADSL interface is between connection attempts.
Down	The ADSL interface is not yet initialized.
No signal from DSLAM	The ADSL interface is not detecting a signal from a DSLAM.

show interface adsl *id* **statistics**

Note: This command is supported beginning with firmware version 8.5.

This command allows you to display statistics for the specified ADSL interface:

Receive frames	Receive octets
Receive errors	Transmit frames
Transmit octets	Transmit errors

---

interface adsl *id* **signaling-mode** { **fdm** | **echo-cancellation** }
show interface adsl *id* **signaling-mode**

These commands allow you to set or show the signalling mode on an ADSL interface. **fdm** = Frequency Division Multiplexing.

---

interface adsl *id* **trellis-coding** { **yes** | **no** }
show interface adsl *id* **trellis-coding**
no interface adsl *id* **trellis-coding**

These commands allow you to set, show, or disable trellis encoding on the specified ADSL interface.

SDSL WAN Interface configuration commands

SDSL WAN Interface Configuration Commands
interface sdsl *id* **clock source** { **internal** | **network** } **show interface sdsl** *id* **clock source** **interface sdsl** *id* **clock rate** *rate-specification* **show interface sdsl** *id* **clock rate** **interface sdsl** *id* **operation mode** { **generic** | **lucent** | **nokia-eoc-fast** | **nokia-fixed** | **paradyne** | **nortel** | **newbridge** } [**default**] **show interface sdsl** *id* **operation mode** **interface sdsl** *id* **pvc** *vpi-value vci-value* **show interface sdsl** *id* **pvc** **interface sdsl** *id* **region** { **annexa** | **annexb** } **show interface sdsl** *id* **region** **interface** { **sdsl** | **isdn** } *id* **rfc1973 dlci** { **16** .. **991** } **show interface** { **sdsl** | **isdn** } *id* **rfc1973 dlci** **interface** { **sdsl** | **isdn** } *id* **rfc1973 enable** { **yes** | **no** } **no interface ethernet** { **sdsl** | **isdn** } *id* **rfc1973 enable** **show interface** { **sdsl** | **isdn** } *id* **rfc1973 enable** **interface** { **sdsl** | **isdn** } *id* **rfc1973 lmi** { **none** | **lmi** | **ccitt** | **ansi** | **annexa** | **annexd** } **no interface** { **sdsl** | **isdn** } *id* **rfc1973 lmi** **show interface** { **sdsl** | **isdn** } *id* **rfc1973 lmi** **show interface sdsl** *id* **status**

interface sdsl *id* **clock source** { **internal** \| **network** }
show interface sdsl *id* **clock source**

These commands allow you to set, change, or show the clock source associated with the SDSL WAN interface.

Note: These commands apply only to frame-based SDSL (R7100) interfaces.

interface sdsl *id* **clock rate** *rate-specification*
show interface sdsl *id* **clock rate**

These commands allow you to set, change, or show the data rate associated with the SDSL WAN interface.

Note: The permissible values for *rate-specification* depend on the type of SDSL WAN interface. For frame-based SDSL (R7100) interfaces, *rate-specification* may be replaced with:

```
{ 160 | 208 | 320 | 416 | 784 | 1040 | 1568 }
```

For cell-based SDSL (R7200) interfaces, *rate-specification* may be replaced with:

```
{ 144...2320 } [{ hunt | locked }]
```

See the table on the next page for possible rate specifications.

Also, **data rate** is accepted as a synonym for **clock rate**.

interface sdsl *id* **operation mode** { **generic** | **lucent** | **nokia-eoc-fast** | **nokia-fixed** | **paradyne** | **nortel** | **newbridge** } [**default**]
show interface sdsl *id* **operation mode**

Note: These commands apply only to ATM-based SDSL interfaces.

If the optional default token is included in the command, various WAN interface parameters will be set to appropriate default values, given the particular mode setting. The parameters and their values are enumerated in the table below. In addition, the data rates accepted by the **interface sdsl** *id* **data rate** command depend on what the operation mode is, and correspond to the values available from the Data Rate pop-up menu on the SDSL Line Configuration screen in the menu console. These acceptable data rates are enumerated below as well.

	Nokia	Lucent	Paradyne	Nortel	Newbridge
VPI	0	0	0	0	0
VCI	38	35	35	38	38
RFC 1483 Mode	Routed	Routed	Routed	Routed	Routed
Data Rate	384k	784k	784k	1536k	2320
Data Rate Mode	HUNT	LOCKED	LOCKED	LOCKED	LOCKED
Clock Source	Network	Network	Network	Network	Network
DLE	rfc1483	rfc1483	rfc1483	rfc1483	rfc1483
Data Rates	192k 384k 768k 1152k 1536k	144k 160k 192k 208k 272k 384k 400k 416k 528k 768k 784k 1040k 1152k 1168k 1536k 1552k 1568k 2320k	144k 272k 400k 528k 784k 1168k 1552k 2320k	144k 160k 192k 208k 272k 384k 400k 416k 528k 768k 784k 1040k 1152k 1168k 1536k 1552k 1568k 2320k	200K 400k 784k 1168k 1552k 2320k

Note that setting the mode value to generic will not change any other WAN interface module parameter; thus, the following command:

```
interface sdsl 1 operation mode generic default
```

will be rejected as a syntax error.

---

interface sdsl *id* **pvc** *vpi-value vci-value*
show interface sdsl *id* **pvc**

These commands allow you to set, change, or show the PVC VPI and VCI values associated with the SDSL WAN interface.

---

interface sdsl *id* **region** { **annexa** | **annexb** }
show interface sdsl *id* **region**

These commands allow you to specify or show the region setting for devices that support multiple (North American/non-North American) regions.

Note: These commands apply only to cell-based SDSL (R7200) interfaces.

---

interface { **sdsl** | **isdn** } *id* **rfc1973 dlci** { **16** .. **991** }
show interface { **sdsl** | **isdn** } *id* **rfc1973 dlci**

These commands allow you to set or show an RFC 1973 DLCI for the SDSL or ISDN WAN interface.

Note that the only WAN interface modules that currently support RFC 1973 are the U/ISDN (31xx) and Copper Mountain SDSL (71xx). Attempts to set or show RFC 1973 parameters on any other WAN interface module will return an error.

---

interface { **sdsl** | **isdn** } *id* **rfc1973 enable** { **yes** | **no** }
no interface { **sdsl** | **isdn** } *id* **rfc1973 enable**
show interface { **sdsl** | **isdn** } *id* **rfc1973 enable**

These commands allow you to enable, disable, or show RFC 1973 (PPP) behavior on the SDSL or ISDN WAN interface.

---

interface { **sdsl** | **isdn** } *id* **rfc1973 lmi** { **none** | **lmi** | **ccitt** | **ansi** | **annexa** | **annexd** }
no interface { **sdsl** | **isdn** } *id* **rfc1973 lmi**
show interface { **sdsl** | **isdn** } *id* **rfc1973 lmi**

These commands allow you to specify, disable, or show the RFC 1973 (PPP) Local Management Interface (LMI) type on the SDSL or ISDN WAN interface.

The keywords **ccitt** and **annexa** are synonyms, as are the keywords **ansi** and **annexd**.

show interface sdsl *id* **status**

This command allows you to display the status of the specified SDSL interface. For a cell-based SDSL (R7200) interface, the possible status strings and their meanings are:

Status String	Meaning
Connected at *xxx* Kbps	The interface is connected to the DSLAM at the specified speed.
Trying *xxx* Kbps	The SDSL interface is attempting to connect to the DSLAM at the specified speed.
Activation Backoff	The SDSL interface is between connection attempts.
Down	The SDSL interface is not yet initialized.
No signal from DSLAM	The SDSL interface is not detecting a signal from a DSLAM.

Priority Queuing (TOS bit) Commands

Priority Queuing Configuration Commands
interface { **adsl** | **sdsl** | **t1** | **serial** } *id* **priority-queuing enable** { **yes** | **no** } **no interface** { **adsl** | **sdsl** | **t1** | **serial** } *id* **priority-queuing enable** **show interface** { **adsl** | **sdsl** | **t1** | **serial** } *id* **priority-queuing enable**

interface { **adsl** \| **sdsl** \| **t1** \| **serial** } *id* **priority-queuing enable** { **yes** \| **no** }
no interface { **adsl** \| **sdsl** \| **t1** \| **serial** } *id* **priority-queuing enable**
show interface { **adsl** \| **sdsl** \| **t1** \| **serial** } *id* **priority-queuing enable**

These commands allow you to enable, disable, or show the priority queuing (TOS) setting for the specified WAN interface.

Differentiated Services (Diffserv) commands

Note: The commands in this section are supported beginning with Firmware Version 8.4.2.

Diffserv Configuration Commands
diffserv enable [**yes** | **no**] **diffserv ratio** [**79 - 100**] **diffserv rule** *id* **name** *string* **diffserv rule** *id* **protocol** [**tcp** | **udp** | **icmp** | **other**] **diffserv rule** *id* **priority** [**off** | **assure** | **expedite** | **reserve**] **diffserv rule** *id* **direction** [**outbound** | **inbound** | **both**] **diffserv rule** *id* **start-port** [**0 - 49151**] **diffserv rule** *id* **end-port** [**0 - 49151**] **diffserv rule** *id* **inside-ip** *x.x.x.x* **diffserv rule** *id* **outside-ip** *y.y.y.y*

diffserv enable [**yes** \| **no**]

This command allows you to enable or disable Differentiated Services (diffserv) for controlling Quality of Service (QoS) queue priority.

diffserv ratio [**79 - 100**]

This command allows you to set the low-high ratio to regulate the level of packets allowed to be pending in the low priority queue.

diffserv rule *id* **name** *string*

This command allows you to create a custom rule specified by *id* with the name *string*. If your applications do not provide Quality of Service (QoS) control, rules allow you to define streams for some protocols, port ranges, and between specific end point addresses.

diffserv rule *id* **protocol** [**tcp** \| **udp** \| **icmp** \| **other**]

This command allows you to specify the protocol for the rule *id*: **tcp**, **udp**, **icmp**, or **other**. **other** is appropriate for rules on protocols with non-standard port definitions. IPSEC and PPTP are common examples. If you specify **other** protocol, you must provide its actual protocol number, with a range of 0 – 255.

diffserv rule *id* **priority** [**off** | **assure** | **expedite**]

This command allows you to specify the priority for the rule *id*: **off**, **assure**, or **expedite.** This is the Quality of Service setting for the rule, based on the TOS bit information. The following table outlines the TOS bit settings and behavior:

QoS Setting	TOS Bit Value	Behavior
off	TOS=000	This custom rule is disabled. You can activate it by selecting one of the two settings below. This setting allows you to pre-define flows without actually activating them.
assure	TOS=001	Use normal queuing and throughput rules, but do not drop packets if possible. Appropriate for applications with no guaranteed delivery mechanism.
expedite	TOS=101	Use minimum delay. Appropriate for VoIP and video applications.

diffserv rule *id* **direction** [**outbound** | **inbound** | **both**]

This command allows you to specify the **direction** of the flow: **outbound**, **inbound**, or **both**. For TCP or UDP protocols, you can optionally specify a range of ports.

diffserv rule *id* **start-port** [**0** - **49151**]

This command allows you to specify the starting port in the range for the rule *id* for TCP or UDP protocols.

diffserv rule *id* **end-port** [**0** - **49151**]

This command allows you to specify the ending port in the range for the rule *id* for TCP or UDP protocols.

diffserv rule *id* **inside-ip** *x.x.x.x*

This command allows you to specify the inside IP address for the rule *id*. For outbound flows, specify an IP address on your LAN. For inbound flows, this setting is ignored.

diffserv rule *id* **outside-ip** *y.y.y.y*

This command allows you to specify the outside IP address for the rule *id*. If you want traffic destined for and originating from a certain WAN IP address to be controlled, enter the IP address here. If you specify all-zeroes, the outside address check is ignored.

For outbound flows, the outside address is the destination IP address for traffic; for inbound packets, the outside address is the source IP address.

PVCs

Note: The commands in this section are supported beginning with firmware release 8.3.1.

PVC Configuration Commands
interface { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **no interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **show interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **tag** *tag* **show interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **tag** **interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **enable** { **yes** | **no** } **no interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **enable** **show interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **enable** **interface adsl** *id* **pvc** { *id* | *tag* } **qos** { **ubr** | **cbr** | **vbr** } **show interface adsl** *id* **pvc** { *id* | *tag* } **qos** **interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **vpi** *vpi-val* **show interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **vpi** **interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **vci** *vci-val* **show interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **vci** **interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **cp** { *profile-id* | *profile-tag* | **default** } **show interface** { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **cp**

interface { **adsl** \| **sdsl** } *id* **pvc** { *id* \| *tag* }
no interface { **adsl** \| **sdsl** } *id* **pvc** { *id* \| *tag* }
show interface { **adsl** \| **sdsl** } *id* **pvc** { *id* \| *tag* }

These commands allow you to set, disable, or show a permanent virtual circuit. You can specify an optional circuit **tag** of up to 14 ASCII characters. The **tag** is used only to identify the circuit for management purposes, and has no significance on the wire; it is merely a convenience to aid in selecting circuits from lists. The default circuit name is “Circuit <n>”, where <n> is replaced with a single decimal ASCII digit (between one and eight) corresponding to the circuit’s position in the list of up to eight circuits.

tag

interface { **adsl** \| **sdsl** } *id* **pvc** { *id* \| *tag* } **tag** *tag*
show interface { **adsl** \| **sdsl** } *id* **pvc** { *id* \| *tag* } **tag**

These commands allow you to set or show a permanent virtual circuit identified by **tag.**

enable

---

interface { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **enable** { **yes** | **no** }
no interface { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **enable**
show interface { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **enable**

These commands allow you to enable, disable, or show a permanent virtual circuit.

---

interface adsl *id* **pvc** { *id* | *tag* } **qos** { **ubr** | **cbr** | **vbr** }
show interface adsl *id* **pvc** { *id* | *tag* } **qos**

Note: These commands are supported beginning with Firmware Version 8.2.

These commands allow you to specify or show the Quality of Service (QoS) type – Unspecified Bit Rate (**ubr**) or Constant Bit rate (**cbr**) – for the specified PVC. *Beginning with Firmware Version 8.3.3*, the **vbr** argument is also supported.

Variable Bit Rate (**vbr**) is characterized by:
- a **pcr** (Peak Cell Rate) value, which is a temporary burst, not a sustained rate, and
- an **scr** (Sustained Cell Rate) value, and
- an **mbs** (Maximum Burst Size/Burst Tolerance) value. **mbs** is the maximum number of cells that can be transmitted at the peak cell rate and should be less than, or equal to the Peak Cell Rate, which should be less than, or equal to the line rate.

VBR has two sub-classes:
a. VBR non-real-time (VBR-nrt): Typical applications are non-real-time traffic, such as IP data traffic. This class yields a fair amount of Cell Delay Variation (CDV).
b. VBR real time (VBR-rt): Typical applications are real-time traffic, such as compressed voice over IP and video conferencing. This class transmits cells with a more tightly bounded Cell Delay Variation. The applications follow CBR.

vpi and vci

---

interface { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **vpi** *vpi-val*
show interface { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **vpi**

These commands allow you to set or show the Virtual Path Identifier value **vpi** for a permanent virtual circuit.

---

interface { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **vci** *vci-val*
show interface { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **vci**

These commands allow you to set or show the Virtual Channel Identifier value **vci** for a permanent virtual circuit.

The **vpi** and **vci** allow you to configure the Virtual Path Identifier and Virtual Channel Identifier which together identify the ATM permanent virtual circuit used between the router and the remote device. The values configured for these items must match those configured in the remote device for data to flow between the devices. The **vpi** may be set to any value between zero (0) and 255.

profile

interface { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **cp** { *profile-id* | *profile-tag* | **default** }
show interface { **adsl** | **sdsl** } *id* **pvc** { *id* | *tag* } **cp**

These commands allow you to set or show the connection profile assigned to the specified PVC.

Note: **default** means that the router will use the first appropriate connection profile or the Default Profile if an appropriate connection profile is not found.

DSL Line Type Interface Configuration Commands

DSL Line Type Interface Configuration Commands
interface dsl *id* **line type** { **g.shdsl** | **sdsl-atm** | **sdsl-hdlc** | **idsl-cmn** | **idsl-leased** | **idsl** } **show interface dsl** *id* **line type**

interface dsl *id* **line type** { **g.shdsl** | **sdsl-atm** | **sdsl-hdlc** | **idsl-cmn** | **idsl-leased** | **idsl** }
show interface dsl *id* **line type**

These commands allow you to set or show the line type for the specified DSL interface.

T1 WAN Interface configuration commands

T1 WAN Interface Configuration Commands
interface t1 *id* **buildout** { **auto** | **0-0.6** | **7.5** | **15.0** | **22.5** } **show interface t1** *id* **buildout**
interface t1 *id* **channels** **count** *integer* [**start** *integer*] [{ **alternating** | **contiguous** }] [**rate** { **56** | **64** | **56k** | **64k** | **Nx56k** | **Nx64k** }] **show interface t1** *id* **channels**
interface t1 *id* **clock source** { **internal** | **network** } **show interface t1** *id* **clock source**
interface t1 *id* **dle** { **ppp** | **hdlc** | **rfc1490** } **show interface t1** *id* **dle**
interface t1 *id* **ds0-autodetect** { **yes** | **no** } **show interface t1** *id* **ds0-autodetect** **no interface t1** *id* **ds0-autodetect**
interface t1 *id* **framing** { **d4** | **esf** } **show interface t1** *id* **framing**
interface t1 *id* **encoding** { **ami** | **b8zs** } **show interface t1** *id* **encoding**
interface t1 *id* **operation line type** { **normal** | **copper-mountain** } **show interface t1** *id* **operation line type**
interface t1 *id* **prm-enable** { **yes** | **no** } **show interface t1** *id* **prm-enable** **no interface t1** *id* **prm-enable**
interface t1 *id* **rfc1973 enable** { **yes** | **no** } **show interface t1** *id* **rfc1973 enable** **no interface t1** *id* **rfc1973 enable**

T1 WAN Interface Configuration Commands (continued)
interface t1 *id* **rfc1973 dlci** { *16..991* } **show interface t1** *id* **rfc1973 dlci** **interface t1** *id* **rfc1973 lmi** { **annexa** | **annexd** | **ansi** | **ccitt** | **lmi** | **none** } **show interface t1** *id* **rfc1973 lmi** **no interface t1** *id* **rfc1973 lmi**

---

interface t1 *id* **buildout** { **auto** | **0-0.6** | **7.5** | **15.0** | **22.5** }
show interface t1 *id* **buildout**

These commands set or display the line buildout for the specified T1 WAN interface.

---

interface t1 *id* **channels**
count *integer*
[**start** *integer*]
[{ **alternating** | **contiguous** }]
[**rate** { **56** | **64** | **56k** | **64k** | **Nx56k** | **Nx64k** }]
show interface t1 *id* **channels**

These commands set or display which DS0 channels are utilized on the specified T1 WAN interface, and the rate of those DS0 channels. The **count** clause is always required. The **start** clause is required unless the **count** clause specifies 24 channels, in which case if the **start** clause is not present, the starting channel number is assumed to be channel 1. If neither the **alternating** nor the **contiguous** keyword is specified, the **contiguous** keyword is assumed unless the line encoding is AMI and the **count** clause specifies two or more channels, in which case the **alternating** keyword is assumed. The **rate** clause is always optional. If the **rate** clause is not present, the value **Nx64k** is assumed, unless the line encoding is AMI, the **count** clause specifies two or more channels, and the **contiguous** keyword is specified, in which case the value **Nx56k** is assumed.

---

interface t1 *id* **clock source** { **internal** | **network** }
show interface t1 *id* **clock source**

These commands set or display the clock source for the specified T1 WAN interface.

---

interface t1 *id* **dle** { **ppp** | **hdlc** | **rfc1490** }
show interface t1 *id* **dle**

These commands set or display the data link encapsulation (DLE) for the specified T1 WAN interface.

Note: **frame-relay** is accepted as a synonym for rfc1490.

---

interface t1 *id* **ds0-autodetect** { **yes** | **no** }
show interface t1 *id* **ds0-autodetect**
no interface t1 *id* **ds0-autodetect**

These commands allow you to set, show, or disable DS0 channel auto-detection on the specified T1 interface.

interface t1 *id* **framing** { **d4** | **esf** }
show interface t1 *id* **framing**

These commands set or display the framing mode for the specified T1 WAN interface.

interface t1 *id* **encoding** { **ami** | **b8zs** }
show interface t1 *id* **encoding**

These commands set or display the line encoding for the specified T1 WAN interface.

Note: If this command changes the line encoding from **b8zs** to **ami** and there are two or more contiguous Nx64k channels in use, the channel data rate will be changed to Nx56k.

interface t1 *id* **operation line type** { **normal** | **copper-mountain** }
show interface t1 *id* **operation line type**

These commands set or display the operation mode for the specified T1 WAN interface. The keyword **copper-mountain** should be specified when connected to a Copper Mountain DSLAM T1 line card; the keyword **normal** should be specified in all other situations.

interface t1 *id* **prm-enable** { **yes** | **no** }
show interface t1 *id* **prm-enable**
no interface t1 *id* **prm-enable**

These commands set or display whether or not ANSI PRMs are sent on the specified T1 WAN interface.

interface t1 *id* **rfc1973 enable** { **yes** | **no** }
show interface t1 *id* **rfc1973 enable**
no interface t1 *id* **rfc1973 enable**

These commands set or display whether or not PPP in Frame Relay (RFC1973) is enabled on the specified T1 WAN interface.

interface t1 *id* **rfc1973 dlci** { *16..991* }
show interface t1 *id* **rfc1973 dlci**

These commands set or display the DLCI used for PPP in Frame Relay (RFC1973) on the specified T1 WAN interface.

interface t1 *id* **rfc1973 lmi** { **annexa** | **annexd** | **ansi** | **ccitt** | **lmi** | **none** }
show interface t1 *id* **rfc1973 lmi**
no interface t1 *id* **rfc1973 lmi**

These commands set or display the Local Management Interface (LMI) type for PPP in Frame Relay (RFC1973) on the specified T1 WAN interface.

T1 Statistic and Diagnostic commands

T1 Statistic and Diagnostic Commands
show interface t1 *id* **errors** { **current** | **interval** *1..96* | **total** }
interface t1 *id* **diagnostic mode** { **local loopback** | **normal** | **remote loopback** | **send** { **all ones** | **blue alarm** | **loopback** } }
show interface t1 *id* **diagnostic mode**
show interface t1 *id* **line status**
show interface t1 *id* **loopback mode**
show interface t1 *id* **loopback status**

---

show interface t1 *id* **errors** { **current** \| **interval** *1..96* \| **total** }

This command displays the error statistics for the specified T1 WAN interface for a particular 15-minute interval during the previous 24-hour period, or the total for the past 24 hours. Specifying the keyword **current** displays the error statistics for the current 15-minute interval. Specifying the keyword **interval** followed by an integer between 1 and 96 displays the error statistics for a prior 15-minute interval. Interval 1 is the most recently completed 15-minute interval, while interval 96 is the interval completed 23 hours and 45 minutes prior to interval 1. Specifying the keyword **total** displays the total error statistics for the last 24 hours.

Example:

```
#show interface t1 1 errors interval 1
15 minutes ending 16:32:44
Errored Seconds           001
Unavailable Seconds       000
Severely Errored Seconds  001
Bursty Errored Seconds    001
Loss of Frame Count       000
Bipolar Violation Count   001

#show interface t1 1 errors total
24 hours ending 16:32:44
Errored Seconds           001
Unavailable Seconds       000
Severely Errored Seconds  001
Bursty Errored Seconds    001
Loss of Frame Count       000
```

```
Bipolar Violation Count   001

#show interface t1 1 errors current
Current Interval elapsed time 02:45
Errored Seconds           002
Unavailable Seconds       000
Severely Errored Seconds  001
Bursty Errored Seconds    001
Loss of Frame Count       000
Bipolar Violation Count   000
```

---

interface t1 *id* **diagnostic mode** { **local loopback** | **normal** | **remote loopback** | **send** { **all ones** | **blue alarm** | **loopback** } }
show interface t1 *id* **diagnostic mode**

This command sets or displays the diagnostic mode for the specified T1 interface. Specifying **local loopback** puts the near end in local payload loopback mode. Specifying **remote loopback** instructs the far end to put itself in payload loopback mode. Specifying **send all ones** or **send blue alarm** (which are synonyms) causes the near end to start sending an all-ones pattern, which puts the far end in the red alarm state, causing it to send back a yellow alarm. Specifying **send loopback** causes the near end to begin sending loopback packets. Specifying **normal** cancels the effect of any previous diagnostic mode command.

After issuing the **diagnostic mode send loopback command**, the loopback progress can be monitored by issuing the **loopback status** command (see below).

show interface t1 *id* **line status**

This command displays the line status on the specified T1 interface. This will display one of the following strings:

```
Red Alarm
Yellow Alarm
Blue Alarm
Normal Operation
```

show interface t1 *id* **loopback mode**

This command displays the loopback mode of the specified T1 interface. This will display one of the following strings:

```
Layer 1 Activation Not Present
Local Payload Loopback Enabled
Remote Line Loopback Enabled
Remote Payload Loopback Enabled
Clear - No Loopback Enabled
```

show interface t1 *id* **loopback status**

This command displays the progress of the loopback test on the specified T1 interface. This will display one of the following strings:

```
Loopback Not Active
PASS (xxxxx good, yyyyy bad packets)
FAIL (xxxxx good, yyyyy bad packets)
```

Examples:

```
#show interface t1 1 loopback status
Loopback Not Active
#interface t1 1 diagnostic mode send loopback
#show interface t1 1 loopback status
PASS (00255 good, 00000 bad packets)
#show interface t1 1 loopback status
FAIL (00000 good, 00256 bad packets)
```

Unprotected Services Configuration Commands

Note: These commands are supported beginning with Firmware Version 8.7.4.

When using an IPSec force-all tunnel, Unprotected Services supports router-generated packets with a source IP address outside the local member range. It works by applying a source address to an internally-generated router *service*, and specifies whether the service should ***not*** be routed by default over the force-all IPSec tunnel.

This permits supporting multiple authentication profles with multiple tunnels, as well as supporting authentication profiles that point to a RADIUS server on the LAN interface. Other applications such as TACACS+, SNMP, syslog, NTP and heartbeat are not forced over the tunnel.

Unprotected Services Configuration Commands
service **interface** [*ip_address* | **cp** | **ethernet**] [*number*] **show** *service* **interface** [**cp** | **ethernet**] [*number*] **no** *service* **interface** *service* **unprotected** [**yes** | **no**] **show** *service* **unprotected** **no** *service* **unprotected**

service **interface** [*ip_address* \| **cp** \| **ethernet**] [*number*]
show *service* **interface** [**cp** \| **ethernet**] [*number*]
no *service* **interface**

These commands allow you to specify, show, or disable the application of a source address to an internally generated router *service*, such that the service should ***not*** be routed by default over a force-all IPSec tunnel.

Applicable internally-generated router *services* are: RADIUS, TACACS+, SNMP, syslog, NTP and heartbeat.

- **interface** specifies from where the traffic is to be sourced.
- For **cp** or **ethernet**, the router will look up its interface address, reducing the chance of error.
- If you enter an *ip_address* that is not a local interface address, the service may either fail to function or the router will override the invalid address. It will then use the interface with a route to the server for the service.

If **no** is used with the commands, the value goes back to the default **0.0.0.0** and **no.**

Note: Only primary Ethernet interfaces are supported; ALANs are not supported.

Examples:

```
remote-server interface 100.110.112.113
remote-server interface cp 3
remote-server interface ethernet 0
remote-server unprotected yes
```

```
remote-server unprotected no
```

---

service **unprotected** [**yes** | **no**]
show *service* **unprotected**
no *service* **unprotected**

These commands allow you to specify, show, or disable whether or not a *service* is "unprotected."

unprotected indicates whether traffic will be sent over a force-all IPSec tunnel or not. All services default to **unprotected no**, meaning that they ***will*** be routed over the IPSec tunnel, unless set to **yes**.

The **no** default enhances security since it requires user intervention to prevent the service from being routed over the IPSec tunnel.

If **no** is used with the commands, the value goes back to the default **0.0.0.0** and **no.**

Examples:

- RADIUS, TACACS+

  The remote-server configuration controls the settings for both TACACS+ and RADIUS servers.

  ```
  remote-server interface 0.0.0.0
  remote-server unprotected no
  ```

- SNMP

  By default, SNMP services use the primary WAN interface, thus no **interface** selection is required.

- syslog

  ```
  system syslog interface 0.0.0.0
  system syslog unprotected no
  ```

- NTP

  ```
  ip ntp interface 0.0.0.0
  ip ntp unprotected no
  ```

- heartbeat

  heartbeat uses the primary WAN interface, thus no **interface** selection is required.

  ```
  heartbeat unprotected no
  ```

If the service fails, a message will be added to the event log. This message has the format;

```
[service] failed. Could not open socket
```

IGMP Configuration Commands

Note: These commands are supported beginning with Firmware Version 8.5.1. IGMP Version 3 is supported beginning with Firmware Version 8.7.

IGMP Configuration Commands
igmp version (**v1** | **v2** | **v3**) **show igmp version**
igmp snooping [**yes** | **no**] **no igmp snooping** **show igmp snooping**
igmp robustness *value* **no igmp robustness** **show igmp robustness**
igmp query-intvl *value* **no query-intvl** **show query-intvl**
igmp query-response-intvl *value* **no query-response-intvl** **show query-response-intvl**
igmp last-member-query-intvl *value* **show igmp last-member-query-intvl**
igmp last-member-query-count *value* **show igmp last-member-query-count**
igmp fast-leave [**yes** | **no**] **no igmp fast-leave** **show igmp fast-leave**
show igmp group
igmp wireless-m2u [**on** | **off**]

igmp version (**v1** \| **v2** \| **v3**)
show igmp version

These commands allow you to set or show the querier's (LAN's) maximum IGMP version that will be used.

Beginning with Firmware version 8.7, **v3** is the default.

---

igmp snooping [**yes** | **no**]
no igmp snooping
show igmp snooping

These commands allow you to enable, disable, or show the status of the Motorola Netopia® Router's ability to "listen in" to IGMP traffic. IGMP "snooping" is a feature of Ethernet layer 2 switches that "listens in" on the IGMP conversation between computers and multicast routers. Through this process, it builds a database of where the multicast routers reside by noting IGMP general queries used in the querier selection process and by listening to other router protocols.

From the host point of view, the snooping function listens at a port level for an IGMP report. The switch then processes the IGMP report and starts forwarding the relevant multicast stream onto the host's port. When the switch receives an IGMP *leave* message, it processes the leave message, and if appropriate stops the multicast stream to that particular port. Basically, customer IGMP messages although processed by the switch are also sent to the multicast routers.

---

igmp robustness *value*
no igmp robustness
show igmp robustness

These commands allow you to specify or show the Motorola Netopia® Router's degree of sensitivity to lost packets. IGMP can recover from robustness minus 1 lost IGMP packet. The default value is 2. The range is 2 – 255.

---

igmp query-intvl *value*
no query-intvl
show query-intvl

These commands allow you to specify or show the amount of time in seconds between IGMP General Query messages sent by the querier router. The default query interval is 125 seconds. The range is 10s – 600s.

---

igmp query-response-intvl *value*
no query-response-intvl
show query-response-intvl

These commands allow you to specify or show the maximum amount of time in tenths of a second that the IGMP router waits to receive a response to a General Query message. The default query response interval is 10 seconds and must be less than the query interval. The range is 5 deci-sec – 255 deci-sec; the default is 100 deci-sec.

---

igmp last-member-query-intvl *value*
show igmp last-member-query-intvl

These commands allow you to specify or show the amount of time in deci-seconds that the router waits to receive a response to a Group-Specific Query message, when **igmp-version** is set to **v2** or **v3**..

---

igmp last-member-query-count *value*
show igmp last-member-query-count

These commands allow you to specify or show the number of Group-Specific Query messages sent.

igmp fast-leave [**yes** | **no**]
no igmp fast-leave
show igmp fast-leave

These commands allow you to specify or show the status of the non-standard procedure **fast-leave** to decrease the time to detect that a group has no more members.

show igmp groups

This command allows you to display the IGMP Snooping table.

igmp wireless-m2u [**on** | **off**]

Note: This command is supported beginning with Firmware Version 8.7.4.

This command allows you enable or disable wireless multicast-to-unicast if **igmp snooping** is set to **yes**.

The router replaces the multicast MAC-address with the physical MAC-address of the wireless client. If there is more than one wireless client interested in the same multicast group, the router will revert to multicasting the stream immediately. When one or more wireless clients leave a group, and the router determines that only a single wireless client is interested in the stream, it will once again unicast the stream.

Global IP Configuration Commands

Global IP Configuration Commands
ip dns { **1** | **2** } *ip-addr* **no ip dns** [{ **1** | **2** } [*ip-addr*]] **show ip dns** [{ **1** | **2** }] **ip domain-name** *string* **no ip domain-name** [*string*] **show ip domain-name** **ip gateway** *ip-addr* **no ip gateway** *ip-addr* **show ip gateway** **backup gateway** *ip-addr* **no backup gateway** *ip-addr* **show backup gateway** **ip ntp period** *value* **no ip ntp period** **show ip ntp period** **ip ntp servers** *ip-addr1 ip-addr2* **no ip ntp servers** **show ip ntp servers** **ip ntp timezone** *value* **no ip ntp timezone** **show ip ntp timezone** **ip route** { *ip-addr/mask-bits* | *ip-addr mask* } *gw-ip-addr* [{ **high** | **low** }] [**advertise** [{**no** | *distance*}] [{**enable** | **disable**}] **no ip route** { *ip-addr/mask-bits* | *ip-addr mask* } *gw-ip-addr* **show ip route** [{ **static** | *ip-addr* | *ip-addr/mask-bits* | *ip-addr mask* }]

Global IP Configuration Commands (continued)

ip state-insp udp-timeout *value*
show ip state-insp udp-timeout

ip state-insp tcp-timeout *value*
show ip state-insp tcp-timeout

ip state-insp dos-detect *value*
show ip state-insp dos-detect

ip state-insp xposed-addr { [*server-list-tag start-ip-addr end-ip-addr*] }
{ [*protocol start-port end-port*] }
no ip state-insp xposed-addr { [*server-list-tag*] }
show ip state-insp xposed-addr *abc*

---

ip dns { **1** | **2** } *ip-addr*
no ip dns [{ **1** | **2** } [*ip-addr*]]
show ip dns [{ **1** | **2** }]

These commands allow you to set, change, delete, or show the router's primary and secondary domain name server addresses.

---

ip domain-name *string*
no ip domain-name [*string*]
show ip domain-name

These commands allow you to set, change, delete, or show the domain name of the router. *string* can be up to 64 characters in length and may contain only valid domain name characters (alpha-numeric characters, dot ("."), and dash ("-"). Note that email addresses contain the at symbol '@' and are not valid domain names.

---

ip gateway *ip-addr*
no ip gateway *ip-addr*
show ip gateway

These commands allow you to set, change, delete, or show the router's default gateway.

---

backup gateway *ip-addr*
no backup gateway *ip-addr*
show backup gateway

This command allows you to set, change, delete, or show the router's default backup gateway.

ip ntp period *value*
no ip ntp period
show ip ntp period

Note: These commands are supported beginning with Firmware Version 8.3.3.

These commands allow you to set, disable, or display the setting of a Network Time Protocol (NTP) server's NTP Update Interval for the router. The *value* should be entered in HHHH:MM format.

ip ntp servers *ip-addr1 ip-addr2*
no ip ntp servers
show ip ntp servers

Note: These commands are supported beginning with Firmware Version 8.3.3.

These commands allow you to set, disable, or display the Host Name or IP Address of a Network Time Protocol (NTP) server you want to specify for the router. You can also specify an alternate NTP server Host Name or IP Address, separated by a space.

ip ntp timezone *value*
no ip ntp timezone
show ip ntp timezone

Note: These commands are supported beginning with Firmware Version 8.3.3.

These commands allow you to set, disable, or display your time zone. You can specify the time zone as + or - Greenwich Mean Time (GMT), or as the standard abbreviation of your zone, if it has one.

For example: GMT -8:00 Pacific Standard Time can be entered as **pst**. The following table shows some standard time settings.

GMT -10:00 Hawaii Standard Time	GMT +3:30 Tehran
GMT -9:00 Alaska Standard Time	GMT +4:00 Russia Zone 3
GMT -8:00 Pacific Standard Time	GMT +4:30 Kabul
GMT -7:00 Mountain Standard Time	GMT +5:00 Russia Zone 4
GMT -6:00 Central Standard Time	GMT +5:30 India
GMT -5:00 Eastern Standard Time	GMT +6:00 Russia Zone 5
GMT -4:00 Atlantic Standard Time	GMT +7:00 Russia Zone 6
GMT -3:30 Newfoundland	GMT +8:00 W. Australia Std. Time
GMT -2:00 Mid-Atlantic Time	GMT +9:00 Japan Standard Time
GMT -1:00 Azores Time	GMT +9:30 Adelaide, Darwin
GMT +0:00 Greenwich Mean Time	GMT +10:00 E. Aust. Std. Time

GMT +1:00 Central Europe Time	GMT +11:00 Russia Zone 10
GMT +2:00 Eastern Europe Time	GMT +12:00 Russia Zone 11
GMT +3:00 Moscow time (MSK)	

ip route { *ip-addr/mask-bits* | *ip-addr mask* } *gw-ip-addr* [{ **high** | **low** }]
[**advertise** [{**no** | *distance*}] [{**enable** | **disable**}]
no ip route { *ip-addr/mask-bits* | *ip-addr mask* } *gw-ip-addr*
show ip route [{ **static** | *ip-addr* | *ip-addr/mask-bits* | *ip-addr mask* }]

The **ip route** and **no ip route** commands allow you to add, change, or delete static routes. The **show ip route static** form of the **show ip route** command displays the configured static routes (including invalid or disabled ones), while the other forms of the **show ip route** command display the router's IP routing table (including any installed (i.e., valid and enabled) static routes).

The destination network may be specified as an IP address and mask in either prefix or dotted-quad notation. *gw-ip-addr* is the IP address of the next-hop router, and should be on one of the router's directly connected IP subnets.

The keywords **high** and **low** control the priority of the static route relative to an identical route learned via RIP. A static route with **high** priority (the default) takes precendence over an identical route learned via RIP, while an identical route learned via RIP takes precedence over a static route with **low** priority.

The keyword **advertise** controls whether or not the router will advertise (redistribute) the static route via RIP. The keyword **advertise** may be followed by a RIP metric (*distance*) between 1 (the default) and 15 inclusive.

If the **show ip route** command includes an optional IP address or IP address and mask, the route, if any, in the IP routing table that pertains to the specified destination network, subnetwork, or host address is displayed.

Examples:

```
ip route 192.168.2.0/24 192.168.1.123 low advertise

no ip route 192.168.2.0 255.255.255.0 192.168.1.123

show ip route

show ip route static
```

Note: Beginning with Firmware Version 8.2, the **ip route** commands will no longer show or set static routes that are default gateways. They will only operate on direct routes (IP address and mask are non-zero). The default gateway routes are now meant to be handled with two commands: 1) **ip gateway**, which configures the Primary Default Gateway, and 2) **backup gateway**, which configures the Backup Default gateway.

DHCP Gen-Options, Option Groups, and Option Filtersets Commands

DHCP Gen-Options commands

DHCP Gen-Options commands
ip dhcp gen-option *tag* **option** *1..255* [**data-type** { **ascii** | **hex** | **dotted-decimal** }] [**data** *data*] **ip dhcp gen-option** *tag* **data-type** { **ascii** | **hex** | **dotted-decimal** } **ip dhcp gen-option** *tag* **data** { *data_of_the_correct_format_given_data-type* } **ip dhcp gen-option** *tag* **priority** { **low** | **high** } **show ip dhcp gen-option** *tag* **priority** { **low** | **high** } **no ip dhcp gen-option** *tag*

ip dhcp gen-option *tag* **option** *1..255* [**data-type** { **ascii** \| **hex** \| **dotted-decimal** }] [**data** *data*]

This command allows you to specify a DHCP generic option set specified by *tag* of one to 15 characters. You can specify up to 20 **gen-option**s. Each can contain up to 100 bytes of data, up to a maximum of 912 bytes of options data total, although, in practical terms, the CLI's 80-character limit will restrict it to fewer. An option specified by a gen-option will be served only if the client requests it.

The following table shows the formats and sizes for known options, and whether or not you can configure a **gen-option** of that type.

Option	Data Format	Data Size (bytes)	Can Configure
0	Empty	0	No
1	IP mask	4	Yes
2	Unsigned 4 byte integer	4	Yes
3 - 11	IP address list	Multiples of 4	Yes
12	String (up to 100 characters)	N	Yes
13	Unsigned 2 byte integer	2	Yes
14 - 15	String (up to 100 characters)	N	Yes
16	Unsigned 4 byte integer	4	Yes
17	String (up to 100 characters)	N	Yes
18	String (up to 100 characters)	N	Yes
19 - 20	Flag	1	Yes
21	IP address & mask list	Multiples of 8	Yes
22	Unsigned 2 byte integer	2	Yes

Option	Data Format	Data Size (bytes)	Can Configure
23	Unsigned 1 byte integer	1	Yes
24	Unsigned 4 byte integer	4	Yes
25	Unsigned 2 byte integer list	Multiples of 2	Yes
26	Unsigned 2 byte integer	2	Yes
27	Flag	1	Yes
28	IP address	4	Yes
29 - 31	Flag	1	Yes
32	IP address	4	Yes
33	IP address and mask list	Multiples of 8	Yes
34	Flag	1	Yes
35	Unsigned 4 byte integer	4	Yes
36	Flag	1	Yes
37	Unsigned 1 byte integer	1	Yes
38	Unsigned 4 byte integer	4	Yes
39	Flag	1	Yes
40	String (up to 100 characters)	N	Yes
41 - 42	IP address list	Multiples of 4	Yes
43	Vendor-specific	String	Yes
44 - 45	IP address list	Multiples of 4	Yes
46	Unsigned 1 byte integer	1	Yes
47	String (up to 100 characters)	N	Yes
48 - 49	IP address list	Multiples of 4	Yes
50	IP address	4	No
51	Unsigned 4 byte integer	4	No
52	Unsigned 1 byte integer	1	No
53	Unsigned 1 byte integer	1	Yes
54	IP address	4	Yes
55	String (up to 100 characters)	N	No
56	String (up to 100 characters)	N	Yes
57	Unsigned 2 byte integer	2	Yes
58 - 59	Unsigned 4 byte integer	4	No
60	String (up to 100 characters)	N	Yes
61	String (up to 100 characters)	N	No
62	String (up to 100 characters)	N	Yes
63	Complex	N	No
64	String (up to 100 characters)	N	Yes

Option	Data Format	Data Size (bytes)	Can Configure
65	IP address list	Multiples of 4	Yes
66 - 67	String (up to 100 characters)	N	Yes
68 - 76	IP address list	Multiples of 4	Yes
77	Pascal string list (length byte + data)	N	Yes
78 - 79	Complex	N	No
80	Empty	0	No
81	Complex	N	No
82	Sub-option list	N	Yes
83	Complex	N	No
84	Undefined	??	Yes
85	IP address list	Multiples of 4	Yes
86 - 87	Unicode String	Multiples of 2	Yes
88	Encoded DN list	N	Yes
89	IP address list	Multiples of 4	Yes
90	Complex	N	No
91 - 97	Undefined/Weakly defined	??	Yes
98	String (up to 100 characters)	N	Yes
99 - 115	Undefined/Weakly defined	??	Yes
116	Flag	1	Yes
117	Unsigned 2 byte integer list	Multiples of 2	Yes
118	IP address	4	Yes
119	Encoded DN list 2	N	Yes
120	Encoded DN list or IPAddress list	N	Yes
121 - 125	Complex	N	No
126 - 127	Undefined	N	Yes
128	IP address list	Multiples of 4	Yes
129 - 223	Undefined/Weakly defined	??	Yes
224 - 254	Private Use	N	Yes
249 (note)	Microsoft uses this instead of 121	N	Yes
255	Empty	0	No

ip dhcp gen-option *tag* **data-type** { **ascii** | **hex** | **dotted-decimal** }

This command allows you to specify the DHCP gen-option data type: **ascii**, **hex** or **dotted-decimal**.

ip dhcp gen-option *tag* **data** { *data_of_the_correct_format_given_data-type* }

This command allows you to specify the **gen-option** data.

- If the **data-type** is **ascii**, then any printable character
- If the **data-type** is **hex**, then an even number of hex characters (e.g. “0123456789AbcdEf”
- If the **data-type** is **dotted-decimal**, then a series of numbers between 0 and 255, separated by a period (.). IP addresses are generally represented in this form.

ip dhcp gen-option *tag* **priority** { **low** | **high** }
show ip dhcp gen-option *tag* **priority** { **low** | **high** }
no ip dhcp gen-option *tag*

These commands allow you to set, display, or disable whether the default handling behavior for a particular option should be overridden by the specified gen-option. For most options this is irrelevant, but at least the following options are handled in the router outside the context of gen-options:

OPTION_SUBNETMASK	1
OPTION_ROUTER	3
OPTION_DNSSERVER	6
OPTION_DOMAINNAME	15
OPTION_BCASTADDR	28
OPTION_ETHERNETENCAP	36
OPTION_NTPSERVERS	42
OPTION_NBNSNAMESERVERS	44
OPTION_NBTCPNODETYPE	46
OPTION_NBTCPSCOPE	47
OPTION_LEASETIME	51
OPTION_RENEWALTIME	58
OPTION_REBINDINGTIME	59
OPTION_VID_VENDOR_SPECIFIC (vendor identity)	125
OPTION_TFTP_SERVER_ADDR	150

Note: In most cases overriding the CPE default behavior is not recommended. Also, if the CPE can, but does not, serve an option in the above list, corresponding low-priority gen-options will be used to serve it.

DHCP Option Groups commands

DHCP Option Groups commands
ip dhcp option-group *tag* [**gen-option** *gen_option_tag*] **show ip dhcp option-group** *tag* **no ip dhcp option-group** *tag* [**gen-option** *gen_option_tag*] **interface ethernet** *id* **address-serve dhcp default-option-group** *option_group_tag* **show interface ethernet** *id* **address-serve dhcp default-option-group** **no interface ethernet** *id* **address-serve dhcp default-option-group**

---

ip dhcp option-group *tag* [**gen-option** *gen_option_tag*]
show ip dhcp option-group *tag*
no ip dhcp option-group *tag* [**gen-option** *gen_option_tag*]

These commands allow you to set, display, or disable one of up to eight DHCP Option Groups. Each Option Group can have a name of between 1 and 15 characters. The name is used in the DHCP filterset syntax to choose what group of gen-options is to be served to a particular DHCP Client.

Option Groups *refer* to **gen-options**; they do not contain them. Deleting a gen-option from an option group does not delete the option. Adding a gen-option to an option-group does not preclude it from being added to another option-group.

---

interface ethernet *id* **address-serve dhcp default-option-group** *option_group_tag*
show interface ethernet *id* **address-serve dhcp default-option-group**
no interface ethernet *id* **address-serve dhcp default-option-group**

These commands allow you to set, display, or disable the option group specified by *option_group_tag* as the default.

DHCP Option Filtersets commands

Support for DHCP option filtering is provided via the filterset settings.

DHCP Option Filtersets commands
ip dhcp-filterset *fs-tag* [**filter** { **new** | **last** | **id** {*1..8*} } [**type** { **dhcp-option** | **hw-address** | **requested-option** }] [**dhcp-option** { *1..255* }] [{ **match-str** *string_w_wildcards ('*' or '?')* } | { [**start-address** *mac-address*] [**end-address** *mac-address*] } }] [**match-action** { **pass** | **discard** | **continue** }] [**match-pool** *ip-addr*] [**match-option-group** { **none** | *opt-group-tag* }] [**absent-action** { **pass** | **discard** | **continue** }] [**absent-pool** *ip-addr*] [**absent-option-group** { **none** | *opt-group-tag* }]] **show ip dhcp-filterset** [*fs-tag* [[**filter**] { **last** | { **id** *1..8* } }]]] **no ip dhcp-filterset** [*fs-tag* [[**filter**] { **last** | {**id** *1..8* } }]]] **interface ethernet** *id* **address-serve dhcp filterset** *fs-tag* **show interface ethernet** *id* **address-serve dhcp filterset** **no interface ethernet** *id* **address-serve dhcp filterset** **bridge-dhcp-filterset** *fs-tag* **show bridge-dhcp-filterset** **no bridge-dhcp-filterset**

---

ip dhcp-filterset *fs-tag* [**filter** { **new** \| **last** \| **id** {*1..8*} }
[**type** { **dhcp-option** \| **hw-address** \| **requested-option** }]
[**dhcp-option** { *1..255* }]
[{ **match-str** *string_w_wildcards ('*' or '?')* } \|
{ [**start-address** *mac-address*] [**end-address** *mac-address*] } }]
[**match-action** { **pass** \| **discard** \| **continue** }]
[**match-pool** *ip-addr*]
[**match-option-group** { **none** \| *opt-group-tag* }]
[**absent-action** { **pass** \| **discard** \| **continue** }]
[**absent-pool** *ip-addr*]
[**absent-option-group** { **none** \| *opt-group-tag* }]]
show ip dhcp-filterset [*fs-tag* [[**filter**] { **last** \| { **id** *1..8* } }]]]
no ip dhcp-filterset [*fs-tag* [[**filter**] { **last** \| {**id** *1..8* } }]]]

These commands allow you to specify, display, or disable a DHCP filterset named *fs-tag*.

A filter can be identified by its ones-based index, [e.g.] 1, or with the special **new** keyword. Subsequent modifications to this filter, assuming no more filters have been added to the filter set yet, must be done by referring to the filter either by **id** (1), or by the other special keyword **last**. Subsequent filters can be added using either **new** or by the next integer filter id. You can always specify the last filter in the set by using **last**. It is an error to attempt to create a new filter whose id is not 1 greater than the id of the last filter.

Up to two filtersets can be added. Your router supports a single LAN DHCP server instance, but an additional filterset is available for use when bridging, to block undesired DHCP traffic. Up to 8 **rules** can be created in the filterset, which are evaluated in order.

- **type:** The rule can either specify an option and option contents **dhcp-option**, a client hardware address range **hw-address**, or an option the client is requesting **requested-option**. By default a rule is of type **dhcp-option**.
- **dhcp-option:** (*1..255*) If the filter is of type **dhcp-option** or **requested-option**, the **dhcp-option** information determines which DHCP option is specified. See page 2-77 for a table of DHCP options.
- **match-str:** The **match-str** *string_w_wildcards* will be compared against the DHCP DISCOVER option data. This string can contain multiple "*" and "?" wildcard substitutions. This is supported only if the filter is of type **dhcp-option**.
- **start-address/end-address:** a client hardware address range. This is supported only if the filter is of type **hw-address**.
- **match-action:**
  - If set to **pass**, the **match-pool** parameter is used to assign a pool start address, or 0.0.0.0 to pass unchanged. DHCP packets matching the option string will pass.
  - If set to **discard**, DHCP packets matching the option string will be blocked.
  - If set to **continue**, the remaining rules in the filter will execute.
- **match-pool:** Specifies the start IP address of the range within a DHCP pool where that range will be used to allocate an address if the wildcard matches.

  The value 0.0.0.0 means regular processing; 255.255.255.255 means discard.
- **match-option-group:** Specifies an option group identified by *opt-group-tag*, or **none** for the **match-action** parameter.
- **absent-action:** As with **match-action**, **absent-action** describes the action taken if the matching DHCP option was not found, using the **absent-pool** parameter to provide the necessary pool start address.

Note: ***absent*** is NOT the opposite of ***match***. The **absent-** part of the rule is taken if the specified option is absent.

- **absent-pool:** Specifies the start IP address of the range within a DHCP pool where that range will be used to allocate an address if the option in the DHCP packet is not present.

  The value 0.0.0.0 means regular processing; 255.255.255.255 means discard.
- **absent-option-group:** Specifies an option group identified by *opt-group-tag*, or **none** for the **absent-action** parameter.

---

interface ethernet *id* **address-serve dhcp filterset** *fs-tag*
show interface ethernet *id* **address-serve dhcp filterset**
no interface ethernet *id* **address-serve dhcp filterset**

These commands allow you to set, display, or disable a DHCP filterset specified by *fs-tag* for the Ethernet interface specified by *id*.

---

bridge-dhcp-filterset *fs-tag*
show bridge-dhcp-filterset
no bridge-dhcp-filterset

These commands allow you to set, display, or disable a DHCP filterset specified by *fs-tag* for use when bridging.

Stateful Inspection Commands

See also:

- "Stateful Inspection Configuration Commands" on page 2-32 for Ethernet interface commands, and
- "Stateful Inspection Commands" on page 3-21 for Connection Profile commands.

Note: The commands in this section are supported beginning with Firmware Version 8.2.

---

ip state-insp udp-timeout *value*
show ip state-insp udp-timeout

These commands allow you to specify or show the UDP timeout value for the stateful inspection feature. The UDP timeout range is between 30 and 65535 seconds.

---

ip state-insp tcp-timeout *value*
show ip state-insp tcp-timeout

These commands allow you to specify or show the TCP timeout value for the stateful inspection feature. The TCP timeout range is between 30 and 65535 seconds.

---

ip state-insp dos-detect [**yes** | **no**]
show ip state-insp dos-detect

Note: These commands are supported beginning with Firmware Version 8.7.

These commands allow you to set or show the status of Denial of Service (DoS) detection in the stateful inspection feature. Packets are monitored for DoS attack detection if this option is set to **yes**. Offending packets maybe discarded if it is determined to be a DoS attack.

---

ip state-insp xposed-addr { [*server-list-tag start-ip-addr end-ip-addr*] }
{ [*protocol start-port end-port*] }
no ip state-insp xposed-addr { [*server-list-tag*] }
show ip state-insp xposed-addr *abc*

These commands allow you to add an entry to the specified list, or, if list does not exist, create the list for the stateful inspection feature. Accepted values for *protocol* are **tcp**, **udp**, or **both**. The **show** command allows you to display exposed entries in the list specified by the tag. The **no** command removes all addresses in the list and deletes the list.

Wireless Configuration Commands

Note: The commands in this section are supported beginning with Firmware Version 8.2 on wireless (802.11)-enabled routers.

Wireless Configuration Commands
wireless enable [**yes** | **no**] **show wireless enable**
wireless closed-system [**yes** | **no**] **show wireless closed-system**
wireless ssid *string* **show wireless ssid**
wireless auto-channel [**off** | **at-startup** | **continuous**] **no wireless auto-channel** **show wireless auto-channel**
wireless tx-power [**full** | **medium** | **fair** | **low** | **minimal**]
show wireless clients
show wireless statistics
wireless default-channel [**no 1**..**14**] **show wireless default-channel**
wireless wep [**enable** | **disable**] **show wireless wep**
wireless default-keyid [**1**..**4**] **show wireless default-keyid**
wireless wep encpt-key [**1**..**4**] *hex_string* **no wireless wep encpt-key** *hex_string* **show wireless wep encpt-key** [**1**..**4**]

Wireless Configuration Commands (continued)
wireless enable [**yes** | **no**] **show wireless enable** **wireless closed-system** [**yes** | **no**] **show wireless closed-system** **wireless ssid** *string* **show wireless ssid** **wireless auto-channel** [**off** | **at-startup** | **continuous**] **no wireless auto-channel** **show wireless auto-channel** **wireless default-channel** [**no 1**..**14**] **show wireless default-channel** **wireless wep** [**enable** | **disable**] **show wireless wep** **wireless default-keyid** [**1**..**4**] **show wireless default-keyid** **wireless wep encpt-key** [**1**..**4**] *hex_string* **no wireless wep encpt-key** *hex_string* **show wireless wep encpt-key** [**1**..**4**]

wireless enable [**yes** \| **no**]
show wireless enable

These commands allow you to enable, disable, or show the status of the wireless option. When disabled, the router will not provide or broadcast any wireless LAN services.

wireless closed-system [**yes** \| **no**]
show wireless closed-system

These commands allow you to enable, disable, or show whether the WLAN is operating as a closed-system. Enabling closed system mode will hide the wireless router's broadcast of the SSID, which prevents its name from appearing on a wireless client when they scan for access points. Therefore, in order to connect, the wireless client would already have to know the SSID. This prevents casual intrusion. Default is **no**.

wireless ssid *string*
show wireless ssid

These commands allow you to specify or show a 32-character string or *Network Name* used to identify this WLAN. Users must select or enter this string on their clients in order to become a part of this WLAN.

wireless auto-channel [**off** | **at-startup** | **continuous**]
no wireless auto-channel
show wireless auto-channel

These commands allow you to set, show, or turn off the wireless autochannel feature (only available for 802.11G models). Autochannel allows the Netopia Router to determine the best channel to broadcast automatically.

Three settings are available: **off**, **at-startup**, and **continuous**.

- **off** is the default setting; the Netopia Router will use the configured default channel.
- **at-startup** causes the Netopia Router at startup to briefly initialize on the default channel, then perform a full two- to three-second scan, and switch to the best channel it can find, remaining on that channel until the next reboot.
- **continuous** performs the at-startup scan, and will continuously monitor the current channel for any other Access Point activity. If Access Point activity is detected on the same channel, the Motorola Netopia® Router will initiate a scan of the other channels, locate a less active one, and switch. Once it has switched, it will remain on this channel for at least 30 minutes before switching again if a new Access Point is detected.

  Note: Channel scans can be disruptive to normal wireless activity and may take a few minutes.

wireless tx-power [**full** | **medium** | **fair** | **low** | **minimal**]

Sets the wireless transmit power, scaling down the router's wireless transmit coverage by lowering its radio power output. Default is **full** power. Transmit power settings are useful in large venues with multiple wireless routers where you want to reuse channels. Since there are only three non-overlapping channels in the 802.11 spectrum, it helps to size the Gateway's cell to match the location. This allows you to install a router to cover a small "hole" without conflicting with other routers nearby.

show wireless clients

This command displays the connected wireless clients, if any.

Example:

```
#show wireless clients
Hardware Address  Status          Privacy    SSID
00-13-ce-62-66-1e Associated      Open       6245 4521
                  48 Mbps, IP: N/A, Tx: 18485, Rx: 0 (bytes)
```

show wireless statistics

This command displays statistics associated with the wireless LAN.

Example:

```
#show wireless statistics

Wireless Statistics
  Wireless Protocol:    IEEE 802.11b/g
  Wireless MAC Addr:    00-00-c5-ca-59-94
  Network ID (SSID):    6245 4521  (Open, No privacy)
  Operating Channel:    6

  Transmit OK           : 16
  Receive OK            : 0
  Tx Errors             : 0
  Rx Errors             : 0
  Rx No Message         : 0
  Rx Octets             : 0
  Rx Unicast Pkts       : 0
  Rx Multicast Pkts     : 0
  Tx Discards           : 0
  Tx Octets             : 5326
  Tx Unicast Pkts       : 0
  Tx Multicast Pkts     : 0

  802.11g Chipset:      Texas Instruments TNETW-1130
  RX Frames:            0
  TX Frames:            0
  Free MSDU:            170  of 170     MSDU Free Watermark:  166
  Free BD:              340 of 340      BD   Free Watermark:  336
  # Out of Packets:     0               # Out of BDs:    0
  # TX Queue Full:      0
```

---

wireless default-channel [**no 1**..**14**]
show wireless default-channel

These commands allow you to specify or show a frequency range within the 2.4Ghz band on which this wireless network will operate. Channel selection depends on government-regulated radio frequencies that vary from region to region. The widest range available is from 1 – 14; in North America, however, only 1 to 11 may be selected. Channel selection can have a significant impact on performance, depending on other wireless activity in proximity to this access point. Channel selection is not necessary at the clients; clients will scan the available channels and look for access points using the same SSID as the client.

---

wireless wep [**enable** | **disable**]
show wireless wep

These commands allow you to enable, disable, or show the status of wireless WEP encryption. When enabled, WEP encryption is used for transmission and reception of wireless data. A single key is selected (see **default-key**) for encryption of outbound/transmitted packets. The WEP-enabled client must have the identical key, of the same length, in the identical slot (1..4) as the access point, in order to successfully receive and decrypt the packet. Similarly, the client also has a 'default' key that it uses to encrypt its transmissions; in order for the access point to receive the client's data, it must likewise have the identical key, of the same length, in the same slot. For simplicity, an access point and its clients need only enter, share, and use the first key.

---

wireless default-keyid [**1**..**4**]
show wireless default-keyid

These commands allow you to specify or show which key is used for encrypting data.

---

wireless wep encpt-key [**1**..**4**] *hex_string*
no wireless wep encpt-key *hex_string*
show wireless wep encpt-key [**1**..**4**]

These commands allow you to specify or show the encryption keys (1 through 4) in hexadecimal. Keys are entered using hexadecimal digits. For 40/64bit encryption, 10 digits are needed; 26 digits for 128bit, and 58 digits for 256bit WEP. Valid hexadecimal characters are 0 – 9, a – f. The **no** command will reset encryption key [*hex_string*] to zero.

Examples:

- 40bit: 02468ACE02
- 128bit: 0123456789ABCDEF0123456789
- 256bit: 592CA140F0A238B0C61AE162F592CA140F0A238B0C61AE162F21A09C

---

wireless mac-auth [**yes** | **no**]
show wireless mac-auth

These commands allow you to enable, disable, or show the status of MAC authentication. Enabling this feature will limit access to the wireless LAN to specific MAC addresses listed with this option. By default, no addresses start in the table.

---

wireless mac-allow *MAC_address*
no wireless mac-allow *MAC_address*
show wireless mac-allow

These commands allow you to specify, show, or reset to zero MAC addresses that will be allowed to participate in the WLAN.

---

wireless mac-deny *MAC_address*
no wireless mac-deny *MAC_address*
show wireless mac-deny

These commands allow you to specify, show, or reset to zero MAC addresses that will be blocked from participating in the WLAN.

wireless mac-delete *MAC_address*

This command allows you to delete the specified MAC address(es) from the table of MAC addresses maintained for this WLAN.

Wireless Privacy Commands (new and revised)

Note: The commands in this section are supported beginning with Firmware Version 8.3.3 on wireless (802.11)- enabled routers.

Wireless Privacy Commands (new and revised)
wireless ssid *string* **no wireless ssid** *string* **show wireless ssid** *string* **wireless privacy** [**off** | **wep-manual** | **wep-auto** | **wpa-psk** | **802.1x**] **show wireless privacy** **wireless psk** *hex_string* **no wireless psk** **show wireless psk** **wireless passphrase** *1..57_character_string* **show wireless passphrase** *1..57_character_string*

wireless ssid *string*
no wireless ssid *string*
show wireless ssid *string*

These commands allow you to specify or show a 32-character string or *Network Name* used to identify this WLAN. Users must select or enter this string on their clients in order to become a part of this WLAN. This is a change to the **wireless essid** command from previous firmware versions. These commands are supported beginning with Firmware Version 8.3.3.

wireless privacy [**off** | **wep-manual** | **wep-auto** | **wpa-psk** | **wpa-802.1x**]
show wireless privacy

These commands allow you to specify, show, or disable the different types of wireless privacy.

Five possible arguments can be set for the **privacy** command:

off	turn off privacy
wep-manual	In this mode you enter encryption keys. Command syntax for entering encryption keys is described in the previous section.
wep-auto	In this mode you enter a passphrase. Command syntax for entering a passphrase is described below.
wpa-psk	In this mode you specify a pre-shared secret key (psk). Command syntax for entering a pre-shared secret key is described below.
wpa-802.1x	In this mode you specify a RADIUS server to be used for user authentication. Command syntax for configuring RADIUS server parameters is described in "RADIUS Authentication Configuration Commands" on page 2-119. Note: The **wpa-802.1x** option is supported beginning with Firmware Version 8.4.2.

wireless psk *hex_string*
no wireless psk
show wireless psk

These commands allow you to specify, disable, or show the Wi-Fi Protected Access (WPA) pre-shared secret key, when wireless privacy is set to **wpa-psk**. The pre-shared secret key can be 8 – 63 alphanumeric characters or 64 hex digits. Entering a blank key will reset the pre-shared key.

wireless passphrase *1..57_character_string*
show wireless passphrase *1..57_character_string*

These commands allow you to set or show a passphrase when wireless privacy is set to **wep-auto**.

Wireless Multiple SSID Commands

Note: These commands are supported beginning with firmware version 8.5.

Wireless Multi-SSID Configuration Commands
wireless block-bridging [**enable** | **disable**] **no wireless block-bridging** **show wireless block-bridging**
wireless multiple-ssid [**enable** | **disable**] **no wireless multiple-ssid** **show wireless multiple-ssid**
wireless first-ssid *string* **show wireless first-ssid**
wireless second-ssid *string* **no wireless second-ssid** **show wireless second-ssid**
wireless third-ssid *string* **no wireless third-ssid** **show wireless third-ssid**
wireless fourth-ssid *string* **no wireless fourth-ssid** **show wireless fourth-ssid**
wireless first-ssid-privacy { **off** | **wep-manual** | **wep-auto** | **wpa-psk** | **wpa-802.1x** } **show wireless first-ssid-privacy**
wireless second-ssid-privacy { **off** | **wep-manual** | **wep-auto** | **wpa-psk** | **wpa-802.1x** } **show wireless second-ssid-privacy**
wireless third-ssid-privacy { **off** | **wep-manual** | **wep-auto** | **wpa-psk** | **wpa-802.1x** } **show wireless third-ssid-privacy**
wireless fourth-ssid-privacy { **off** | **wep-manual** | **wep-auto** | **wpa-psk** | **wpa-802.1x** } **show wireless fourth-ssid-privacy**

Wireless Multi-SSID Configuration Commands (continued)
wireless first-ssid-wpaver [**all** | **WPA-v1-only** | **WPA-v2-only**] **show wireless first-ssid-wpaver** **wireless second-ssid-wpaver** [**all** | **WPA-v1-only** | **WPA-v2-only**] **show wireless second-ssid-wpaver** **wireless third-ssid-wpaver** [**all** | **WPA-v1-only** | **WPA-v2-only**] **show wireless third-ssid-wpaver** **wireless fourth-ssid-wpaver** [**all** | **WPA-v1-only** | **WPA-v2-only**] **show wireless fourth-ssid-wpaver** **wireless second-ssid-psk** *string* **wireless third-ssid-psk** *string* **wireless fourth-ssid-psk** *string*

---

wireless block-bridging [**enable** \| **disable**]
no wireless block-bridging
show wireless block-bridging

These commands allow you to block, unblock, or show the status of wireless to wireless inter-client communication.

---

wireless multiple-ssid [**enable** \| **disable**]
no wireless multiple-ssid
show wireless multiple-ssid

These commands allow you to enable, disable, or show the status of multiple wireless SSIDs. For the second, third, and fourth SSIDs, you must first set **multiple-ssid** to **enable**. The **no** command will reset multiple SSIDs to disabled.

---

wireless first-ssid *string*
show wireless first-ssid

These commands allow you to specify or show a 32-character string or *Network Name* used to identify this WLAN. Users must select or enter this string on their clients in order to become a part of this WLAN. The **first-ssid** command is also functionally equivalent to the **wireless ssid** command. See "Wireless Privacy Commands (new and revised)" on page 2-91.

---

wireless second-ssid *string*
no wireless second-ssid
show wireless second-ssid

These commands allow you to specify or show a 32-character string or *Network Name* used to identify a second SSID. This SSID will not be broadcasted. The **no** command will reset the second SSID to empty.

---

wireless third-ssid *string*
no wireless third-ssid
show wireless third-ssid

These commands allow you to specify or show a 32-character string or *Network Name* used to identify a third SSID. This SSID will not be broadcasted. The **no** command will reset the third SSID to empty.

---

wireless fourth-ssid *string*
no wireless fourth-ssid
show wireless fourth-ssid

These commands allow you to specify or show a 32-character string or *Network Name* used to identify a fourth SSID. This SSID will not be broadcasted. The **no** command will reset the fourth SSID to empty.

---

wireless first-ssid-privacy { **off** | **wep-manual** | **wep-auto** | **wpa-psk** | **wpa-802.1x** }
show wireless first-ssid-privacy

These commands allow you to specify or show the privacy setting for the first, and primary, SSID. Setting the privacy setting of this SSID will reset all of the SSID privacy settings to this value. If you want to set the primary to **wpa-802.1x** and one of the other SSIDs to **wpa-psk**, you must set the primary's first.

---

wireless second-ssid-privacy { **off** | **wep-manual** | **wep-auto** | **wpa-psk** | **wpa-802.1x** }
show wireless second-ssid-privacy

wireless third-ssid-privacy { **off** | **wep-manual** | **wep-auto** | **wpa-psk** | **wpa-802.1x** }
show wireless third-ssid-privacy

wireless fourth-ssid-privacy { **off** | **wep-manual** | **wep-auto** | **wpa-psk** | **wpa-802.1x** }
show wireless fourth-ssid-privacy

Note: Wireless privacy commands for multiple SSIDs are supported beginning with Firmware Version 8.5.1.

These commands allow you to specify or show a privacy setting for each SSID. The privacy setting must match the **first-ssid-privacy** setting, unless **first-ssid-privacy** is **wpa-802.1x**. In that case, subsequent SSIDs may be set to **wpa-psk**, if desired.

wireless first-ssid-wpaver [**all** | **WPA-v1-only** | **WPA-v2-only**]
show wireless first-ssid-wpaver

wireless second-ssid-wpaver [**all** | **WPA-v1-only** | **WPA-v2-only**]
show wireless second-ssid-wpaver

wireless third-ssid-wpaver [**all** | **WPA-v1-only** | **WPA-v2-only**]
show wireless third-ssid-wpaver

wireless fourth-ssid-wpaver [**all** | **WPA-v1-only** | **WPA-v2-only**]
show wireless fourth-ssid-wpaver

Note: WPA privacy version commands are supported beginning with Firmware Version 8.6.

These commands allow you to set or show the first, second, third, or fourth SSID's allowed WPA versions if WPA is selected as a privacy option for that SSID. The default is **all**.

wireless first-ssid-psk *string*
wireless second-ssid-psk *string*
wireless third-ssid-psk *string*
wireless fourth-ssid-psk *string*

These commands allow you to enter a WPA preshared key string for the specified SSID, if wireless privacy is set to **wpa-psk** for that SSID.

Wireless MultiMedia (WMM) Configuration Commands

Note: These commands are supported beginning with Firmware Version 8.7.

Wireless MultiMedia (WMM) Configuration Commands
wireless wmm { **off** | **diffserv** } **show wireless wmm**

wireless wmm { **off** | **diffserv** }
show wireless wmm

These commands allow you to set or show the status of the wireless multimedia option. Wireless Multimedia currently implements wireless Quality of Service (QoS) by transmitting data depending on **diffserv** priority settings.

ARP Configuration Commands

ARP Configuration Commands
arp *ip-addr hw-address* **arp** *ip-addr hw-address interface-id* (D7100 CSU only) **no arp** *ip-addr hw-address* **show arp static** **clear arp-cache** **show arp-cache**

Note: The *hw-address* format for an Ethernet MAC address is six hexadecimal values between 00 and FF inclusive separated by colons. Thus, the following is an example of a valid Ethernet MAC address *hw-address*:

```
00:00:C5:70:00:04
```

arp *ip-addr hw-address*
arp *ip-addr hw-address interface-id* (D7100 CSU only)

This command allows you to create or modify a global ARP cache entry. If the model number of the Motorola Netopia® router is D7100 (CSU), then the interface-id is required so that the device knows the interface with which to associate the ARP entry.

no arp *ip-addr hw-address*

This command allows you to remove a global ARP cache entry. Note that this does not affect entries in the interface-specific caches acquired via ARP requests and responses. To flush the interface-specific ARP caches, use the **clear arp-cache** command, described below.

show arp static

This command displays global ARP cache entries configured using the **arp** command described above.

clear arp-cache

This command allows you to flush all of the interface-specific ARP caches. It does not affect any entries in the global ARP cache, described above.

show arp-cache

This command allows you to display the global ARP cache as well as the ARP cache for each active interface that supports ARP.

ARP and Bridge timeout settings

Note: These commands are supported beginning with Firmware Version 8.4.2.

ARP and Bridge Timeout Settings
ip arp-timeout *timeout* **system bridge-timeout** *timeout*

ip arp-timeout *timeout*

This command allows you to set the *timeout* value for ARP timeout. Default = 600 secs (10 mins); range = 60 secs - 6000 secs (1–100 mins).

system bridge-timeout *timeout*

This command allows you to set the *timeout* value for bridging table timeout. Default = 30 secs; range = 30 secs – 6000 secs (1–100 mins).

Scheduled Connections Configuration Commands

Note: Commands in this section are supported beginning with Firmware Version 8.3.1.

Scheduled Connections Configuration Commands
schedule *id* **enable** (**yes** | **no**) **schedule** *id* **frequency** (**weekly** | **once**) **schedule** *id* **type** (**force-up** | **force-down** | **demand-allowed** | **demand-block** | **periodic** | **random-retry**) **schedule** *id* **periodic interval** (**15min** | **30min** | **1hour** | **2hour** | **3hour** | **4hour** | **4hour** | **8hour**) **schedule** *id* **random interval** { *xxx* } **schedule** *id* **cp** *cp_name* **schedule** *id* **date** (**sunday** | **monday** | **tuesday** | **wednesday** | **thursday** | **friday** | **saturday** | **weekdays** | **weekends** | **everyday**) **schedule** *id* **start time** *HH:MM* (**am** | **pm**)

---

schedule *id* **enable** (**yes** \| **no**)

This command allows you to enable or disable the specified Scheduled Connection.

---

schedule *id* **frequency** (**weekly** \| **once**)

This command allows you to specify whether the Scheduled Connection is to be invoked weekly or only once.

---

schedule *id* **type** (**force-up** \| **force-down** \| **demand-allowed** \| **demand-block** \| **periodic** \| **random-retry**)

This command allows you to specify the Scheduled Connection type:

- **force-up** – this connection will be maintained whether or not there is a demand call on the line.
- **force-down** – this connection will be torn down or blocked whether or not there is a demand call on the line.
- **demand-allowed** – this schedule will permit a demand call on the line.
- **demand-block** – this schedule will prevent a demand call on the line.
- **periodic** – the connection is retried several times during the scheduled time.
- **random-retry** – operates as follows:

  First, it will wait 0 to 60 seconds before starting, then it will try three times to bring the connection up as quickly as possible;

  Second, on each successive retry after these first three attempts it will wait a random number of seconds between zero and a user-specified maximum.

  Should the connection come up, and subsequently go down, the Scheduled Connection will start over with three retries. Switched connections have a variable redial back-off time depending on the interface type. Consequently, the first three attempts for such connections will be slower. Once the connection is up it will be forced to remain up.

schedule *id* **periodic interval** (**15min** | **30min** | **1hour** | **2hour** | **3hour** | **4hour** | **4hour** | **8hour**)

If you specify **periodic**, this command allows you to specify for how long the system will retry to bring up the connection.

schedule *id* **random interval** { *xxx* }

If you specify **random-retry**, this command allows you to set the upper limit for the number of minutes to use for the retry time (the attempts after the first three attempts). It accepts values of 1 – 255 minutes; the default setting is 5 minutes. With a setting of 5 minutes it will try every 0 – 300 seconds after the first three retries to bring up the connection.

schedule *id* **cp** *cp_name*

This command allows you to specify the Connection Profile to be used by this Scheduled Connection to connect.

schedule *id* **date** (**sunday** | **monday** | **tuesday** | **wednesday** | **thursday** | **friday** | **saturday** | **weekdays** | **weekends** | **everyday**)

This command allows you to specify which days of the week the Scheduled Connection will be invoked.

schedule *id* **start time** *HH:MM* (**am** | **pm**)

This command allows you to set the starting time, and whether AM or PM, for the specified Scheduled Connection. You must enter the time in the format HH:MM, where HH is a one- or two-digit number representing the hour and MM is a one- or two-digit number representing the minutes. The colon is mandatory. For example, the entry 1:3 (or 1:03) would be accepted as 3 minutes after one o'clock. The entry 7:0 (or 7:00) would be accepted as seven o'clock, exactly. The entries 44, :5, and 2: would be rejected.

Default Profile Configuration Commands

Default Profile Configuration Commands
dp ip dhcp client mode { **standard** | **copper-mountain** | **cmn** } **show ip dhcp client mode**

dp ip dhcp client mode { **standard** | **copper-mountain** | **cmn** }
show ip dhcp client mode

These commands allow you to set or show the router's Default Profile DHCP mode, whether **standard**, **copper-mountain**, or **cmn**.

The default profile, and IP configuration structures include a **dhcp client mode** setting that selects between the **standard** RFC 2131 standards-based mode of operation (the default), and the **copper-mountain** or **cmn** proprietary mode of operation.

When the DHCP client is activated on a RFC1483 MER interface, it examines the **dhcp client mode** in the associated connection profile (or the default profile there was no explicitly configured connection profile). If the **dhcp client mode** specifies **standard**, the DHCP client initializes the htype and hlen fields in the header of its DHCP requests to the appropriate values for an RFC1483 MER interface (htype = 1 and hlen = 6). If the **dhcp client mode** specifies **copper-mountain** or **cmn**, the DHCP client initializes the htype and hlen fields in the header of its DHCP requests to zero.

When the DHCP client is activated on an Ethernet WAN interface, it examines the **dhcp client mode** in the associated IP configuration structure, and behaves as described above for the RFC1483 MER DHCP client.

Note: **cmn** is accepted as a synonym for **copper-mountain**.

Both of these commands are supported only on hardware that has a copper mountain SDSL, IDSL, or Ethernet WAN interface!

Frame Relay Configuration Commands

Frame Relay Configuration Commands
frame-relay dlci *number* [**tag** *tag*] [**ip-addr** { 0.0.0.0 | *remote-ip-addr* }] [**cir** { **default** | *1-accessrate* }] [**bc** { **default** | *1-accessrate* }] [**be** { **default** | *0-accessrate* }] [{**disable** | **enable**}] **no frame-relay dlci** *number*
frame-relay lmi type { **none** | **annexa** | **annexd** | **ansi** | **ccitt** | **lmi** } **no frame-relay lmi type** **show frame-relay lmi type**
frame-relay tim { **none** | **standard** | **buffered** }
show frame-relay lmi statistics
show frame-relay pvc

frame-relay dlci *number* [**tag** *tag*] [**ip-addr** { 0.0.0.0 \| *remote-ip-addr* }]
[**cir** { **default** \| *1-accessrate* }] [**bc** { **default** \| *1-accessrate* }] [**be** { **default** \| *0-accessrate* }]
[{**disable** \| **enable**}]

This command allows you to manually configure a DLCI. If the IP address is 0.0.0.0 the router will attempt to auto-discover the remote IP address. The router supplies a default DLCI 16 with a 0.0.0.0 address, so in many cases manually configuring a DLCI is unnecessary.

Examples:

To add a DLCI using all of the default values, type:

```
frame-relay dlci 17
```

The tag (i.e., name) of the DLCI defaults, in this case, to "DLCI 17". To specify a different name, type:

```
frame-relay dlci 17 tag "My DLCI"
```

For descriptions of the other parameters available for configuration (and their default values) see the *Motorola Netopia® Router User's Reference Guide*.

---

no frame-relay dlci *number*

This command deletes the DLCI identified by *number*.

---

frame-relay lmi type { **none** | **annexa** | **annexd** | **ansi** | **ccitt** | **lmi** }
no frame-relay lmi type
show frame-relay lmi type

These commands allow you to change or display the Frame Relay Local Management Interface (LMI) type. The keywords **ccitt** and **annexa** are synonyms, as are the keywords **ansi** and **annexd**.

---

frame-relay tim { **none** | **standard** | **buffered** }

This command allows you to set the routers's Frame Relay transmit injection management **tim**.

---

show frame-relay lmi statistics

This command displays Frame Relay Local Management Interface (LMI) statistics.

Example:

```
#show frame-relay lmi statistics
LMI Tx Status Enquiries:          10
LMI Rx Status Responses:          10
LMI Timeouts:                      0
LMI Errors:                        0
LMI Failures:                      0
```

---

show frame-relay pvc

This command displays the status of each Frame Relay permanent virtual circuit (pvc).

Example:

```
#show frame-relay pvc
DLCI:    16, status: active
DLCI:    17, status: active
```

Miscellaneous Commands

Miscellaneous Commands
clear **ping** [**ip**] { *ip-addr* | *hostname* } [**count** *count*] [**timeout** *milliseconds*] [**delay** *milliseconds*] [**size** *bytes*] [**source** *ip-addr*] **ping oam interface sdsl** *id* **pvc** {*id* | *tag* | *vpi/vci* } [**count** *count*] [**timeout** *seconds*] **receive tftp config** [*server-name file-name*] **receive tftp html** [*server-name file-name*] [**noreboot**] **receive tftp** [**wan** { **1** | **2** }] **firmware** [*server-name* [*file-name*]] **send tftp config** [*server-name file-name*] **show tftp last error** **show tftp status** **receive xmodem** [**wan** { **1** | **2** }] **firmware** **show xmodem status** **reset** [**factory** | **bridge** | **router**] **show config** **show history** **show memory** **show model** **show system information** **[show] version** [**cli**] [**firmware**] [**hardware**] [**mib**] [**wan 1**] [**wan 2**] **system restart-delay** [*time (minutes)* | **no**] **show system restart-delay** **traceroute** [*hostname* | *ip-address*] **upgrade** *key-value* **upnp enable** [{**yes** | **no**}] **no upnp enable** **show upnp enable**

clear

This command erases the screen.

ping [**ip**] { *ip-addr* | *hostname* } [**count** *count*] [**timeout** *milliseconds*] [**delay** *milliseconds*] [**size** *bytes*] [**source** *ip-addr*]

The **ping** command allows you to send ICMP echo requests to another network device. You can specify the destination using either an IP address in dotted-quad notation or a hostname. The default **count** is 5, the default **timeout** and **delay** are 1000 milliseconds (1 second) each, and the default **size** is 100 bytes. The **size** value you specify includes the size of the IP and ICMP packet headers. The minimum **size** value is 28 and the maximum **size** value is 1664. **source** specifies which router interface IP address *ip-addr* to use as the source IP address. If the ping goes out through an interface that has NAT (Network Address Translation) enabled, then the source address will be translated.

For each timely ICMP echo response received, an exclamation point ("!") is displayed; for each timeout, a period (".") is displayed; and for each ICMP destination unreachable received, an uppercase letter U ("U") is displayed. When the ping operation completes, statistics are displayed including the total number of ICMP echo requests sent and ICMP echo responses received, the success rate as a percentage, and the minimum, average, and maximum round trip times.

To abort a ping operation while it is in progress, type Control-C.

Example:

```
#ping www.netopia.com
Translating "www.netopia.com"... (163.176.4.31) [OK]
Type Control-C to abort.
Sending 5, 100-byte ICMP Echos to 163.176.4.32, timeout is 1 second:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/4/10 ms
#
```

ping oam interface sdsl *id* **pvc** {*id* | *tag* | *vpi/vci* } [**count** *count*] [**timeout** *seconds*]

This command initiates a process by which up to *n* (*count*) OAM loopback tests are performed at *n* (*seconds*) delay between each.

For each individual OAM loopback test failure, a period '.' will be output. For each individual OAM loopback test success, an exclamation mark '!' will be output. A running total of successes and failures will be maintained for the duration of the test. The success-to-failure rate is output at the conclusion. If any OAM loopback test fails for a reason other than a timeout, the command will be aborted immediately.

Example:

```
ping oam interface sdsl 1 pvc count 17

Type Control-C to abort.

Performing 17 OAM loopback tests; timeout is 5 seconds.

.....

Success rate is 0 percent (0/17)
```

The default count is 5, and the default timeout is 0.

---

receive tftp config [*server-name file-name*]

This command allows you to initiate a configuration file upload from the command line interface. If the TFTP server name and config file name were set previously, either by a previous invocation of this command or via the menu console, then you do not need to supply the server and file name. If the upload is successful, the router will automatically reboot. If the upload is unsuccessful, in menu mode you can go to:

for information about what went wrong.

---

receive tftp [**wan** { **1** | **2** }] **firmware** [*server-name* [*file-name*]]

This command allows you to receive a firmware file that is located remotely on a TFTP server into the router by means of a TFTP file transfer.

The **wan 1** and **wan 2** options allow you to specify that the firmware upgrade is to be sent one of the WAN interface modules. As with the regular **receive tftp firmware** comand, *server-name* and *file-name* are optional if they are already specified. If *server-name* or *file-name* are not already specified, it is necessary to specify them. (Specifying them with this command will set them.)

---

send tftp config [*server-name file-name*]

The configuration file will be stored on the TFTP server as a binary image. Note that the file must already exist on the server and be writable.

---

show tftp last error

An empty string indicates no error. Other possible return values are: File not found, Unauthorized, Disk Full, Illegal Operation, Bad TID, File Already Exists, No Such User, and Unknown Error.

---

show tftp status

The possible return values are: Idle, Reading Firmware, Reading Wanlet Firmware, Reading Config, Writing Config, Idle, **TRANSFER FAILED**.

---

receive xmodem [**wan** { **1** | **2** }] **firmware**

This command allows you to receive a firmware file that is located locally into the router by means of an XMODEM file transfer.

The **wan 1** and **wan 2** options allow you to specify that the firmware upgrade is to be sent one of the WAN interface modules.

The underlying routine that is invoked blocks until either the file has been successfully received or some condition caused the transfer to abort. Therefore when this command is invoked the following will appear:

```
#receive xmodem firmware
This will take a long time, and the console will be inactive during this time.
If the transfer is successful, the router will reboot automatically.
Press Ctl-C to abort, or...
Press Return/Enter to continue; you will have 10 seconds to start the transfer.
```

Once you press Return the console is inactive until the XMODEM routine returns. If the transfer is unsuccessful, the following message displays:

```
XModem transfer Failed!
#
```

---

show xmodem status

This command displays the status of an xmodem firmware file transfer. Possible values are:

```
- Reading Wanlet Firmware
- Sending firmware to wanlet
- Sending block x of y to wanlet
- Validating Firmware
- Decompressing Firmware
- Checking CRC
- Updating Firmware: x% complete
- **Upgrade SUCCESSFUL**
- Insufficient Memory
- **Upgrade FAILED**
```

---

reset [**factory** | **bridge** | **router**]

This command allows you to reset the router from the command line interface. If the optional keyword **factory** is specified, the parameters of the router are reset to their initial, factory default values. Adding the optional keyword **bridge** or **router** will switch the device to bridge or router mode. This command must be completely typed out; it may not be abbreviated.

WARNING: In many cases a factory reset will cause you to be unable to communicate with the router over the network, since the Ethernet LAN IP address of the router will revert to its default value, which for most models is 192.168.1.1. Consult with your service provider to be sure of your default router IP address.

Similarly, switching between bridge and router mode can also cause loss of communication with the device.

---

show config

This command displays the entire list of current configuration settings of the device.

show history

This command displays the command history buffer, which contains a record of the commands that have been entered on the console. The list is limited to the most recent ten commands entered. The oldest command appears first, and the most recent command appears last.

show memory

This command displays the system memory status.

Example:

```
#show memory
Memory status:
Heap: total bytes 7936960 (free 4437424, allocated 3499536)
System Packet Buffers: total 1000 free 993 min 992
Image usage: text 3931180, data 536336, bss 370412
Heap status: peak usage 3520512, lowest free 4416448, fragments 6
```

show model

This command displays the device's model number.

Example:

```
#show model
Netopia R7200 Router
```

show system information

This command shows the same information as is shown in the System Information screen in the console menu user interface:

- Serial Number
- Firmware Version
- Processor Speed (MHz)
- Flash ROM Capacity (MBytes)
- DRAM Capacity (MBytes)
- Ethernet
- WAN Interface

show version [**cli**] [**firmware**] [**hardware**] [**mib**] [**wan 1**] [**wan 2**]

This command allows you to display some or all of the router's version strings. If you don't specify anything after the keyword **version**, the router displays a list of all of the version strings; otherwise it displays the specified version string.

The R-Series WAN interface module firmware version is displayed for the specified WAN module, **wan 1** or **wan 2**. **wan1** is an acceptable substitute for **wan 1**; **wan2** is an acceptable substitute for **wan 2**.

The CLI, hardware, and product-specific SNMP MIB version strings consist of a major version, minor version, release stage, and revision, displayed in the (decimal) form “MM.mmsrr”, where:

‘MM’ is the major version,

‘mm’ is the minor version,

‘s’ is the release stage — e (experimental), d (development), a (alpha), b (beta), or f (final),

‘rr’ is the revision.

The firmware version string consists of a major version, minor version, point release version, release stage, and revision, displayed in the (decimal) form “MM.mm.ppsrr”, where:

‘MM’ is the major version,

‘mm’ is the minor version,

‘pp’ is the point release version,

‘s’ is the release stage – d (development), a (alpha), b (beta), fc (final candidate), or f (final),

‘rr’ is the revision.

Examples:

```
#show version
cli      version:  01.00d00
firmware version:  04.10.00
hardware version:  01.00f00
mib      version:  01.00f00
wan1     version:  fw v1.0.6
wan1     version:  V2.210-N-V90_2M_DLS
#
```

system restart-delay [*time (minutes)* | **no**]
show system restart-delay

Note: These commands are supported beginning with firmware version 8.5.

The **system restart-delay** command schedules a system reset for a specified number of minutes from now. The **show** command displays how many minutes are left before a reset will take place. The **no** modifier, or setting the time to zero, cancels the scheduled reset.

traceroute [*hostname* | *ip-address*]

Note: This command is supported beginning with firmware version 8.5.

This command allows you to display the path to a destination by showing the number of hops and the router addresses of these hops for each router encountered between the Motorola Netopia® Router and a destination hostname or IP address. Firewalls or other security measures that block PING traffic also block traceroutes.

upgrade *key-value*

If the upgrade operation is successful the router will reboot. Otherwise, an error will be returned. The *key-value* argument is the upgrade key that was received when the upgrade was purchased.

upnp enable [{**yes** | **no**}]
no upnp enable
show upnp enable

Note: These commands are supported beginning with Firmware Version 8.3.3.

These commands allow you to enable, disable, or show the status of Universal Plug and Play. UPnP™ is a set of protocols that allows a PC to automatically discover other UPnP devices (anything from an internet gateway device to a light switch), retrieve an XML description of the device and its services, control the device, and subscribe to real-time event notification.

By default, **upnp** is enabled on the Motorola Netopia® Gateway.

For Windows XP users, the automatic discovery feature places an icon representing the Motorola Netopia® Gateway automatically in the “My Network Places” folder.

PCs using UPnP can retrieve the Gateway’s WAN IP address, and automatically create NAT port maps. This means that applications that support UPnP, and are used with a UPnP-enabled Motorola Netopia® Gateway, will not need application layer gateway support on the Motorola Netopia® Gateway to work through NAT.

You must reboot the Motorola Netopia® device for this setting to take effect.

IP Network Address Translation (NAT) Commands

IP Network Address Translation (NAT) Commands
ip nat public *tag* **dynamic** *from-address to-address* **ip nat public** *tag* **static** *pub-from-address pub-to-address* **ip nat public** *tag* **pat** { *pub-address* | 0.0.0.0 } [*from-port to-port*] **no ip nat public** [*tag*] **ip nat map** *list-tag priv-from-address priv-to-address pool-tag* **no ip nat map** [*list-tag*] **show ip nat map** *list-tag* **ip nat server** *list-tag priv-ip-addr pub-ip-addr* { *port-name pub-start-port* [*pub-end-port*] } { both | tcp | udp } *priv-start-port* **no ip nat server** [*list-tag*] **show ip nat server** *list-tag* **show ip nat translation**

Overview

Network Address Translation (NAT) makes use of five basic structures:

- "public" or externally visible address ranges
- "map" rules that bind an interior, private address range with a public address range
- "map-lists", which are ordered lists of maps
- "servers", which define a mapping between a private IP address and port and a public IP address (and the same port)
- "server-lists" which are lists of servers.

ip nat public *tag* **dynamic** *from-address to-address*

This command allows you to allocate a dynamic range of exterior, public addresses for use by Network Address Translation. This range of addresses will be associated dynamically with private addresses you will define when you create a map. *tag* is the name you assign to the range, and it can contain up to 16 characters.

Example:

The following creates a dynamic public range of 8 addresses starting at 163.176.12.1:

```
ip nat public "my second range" dynamic 163.176.12.1 163.176.12.8
```

ip nat public *tag* **static** *pub-from-address pub-to-address*

This command allows you to allocate a range of exterior, public addresses for use by Network Address Translation. This range of addresses will be associated one-to-one with private addresses you will define when you create a map, described later. *tag* is the name you assign to the range, and it can contain up to 16 characters. The address range is defined to be from the first address to the last address, inclusive.

Example:

The following creates a static public range of 8 addresses starting at 163.176.12.1:

```
ip nat public "my first range" static 163.176.12.1 163.176.12.8
```

ip nat public *tag* **pat** { *pub-address* | 0.0.0.0 } [*from-port to-port*]

This command allows you to configure a PAT public range. Since PAT allows you to map multiple private addresses to a single public address, you specify only the public address. Optionally you can specify a range of ports to be used by PAT. The lowest allowed port is 1025, and the highest allowed port is 65535. If you do not specify the port range explicitly the default range is 49152 to 65535 inclusive.

If you specify 0.0.0.0 as the public address, whatever address is negotiated by PPP or DHCP when the WAN connection is established, this pool will adopt. Since neither PPP nor DHCP are capable of assigning more than one address to a single client you should have at most one (active) PAT public range whose public address is 0.0.0.0.

Example:

The following command creates a PAT public range at 163.176.12.1, with a port range from 10000 to 65535:

```
ip nat public "my pat range" pat 163.176.12.1 10000 65535
```

no ip nat public [*tag*]

This command allows you to delete a public range whose name is *tag*. Since tags must be unique this command works for both static and PAT public ranges. If the *tag* is omitted then all public ranges will be deleted.

Note: Use of this command can have significant side effects, because any map in any map list that refers to a public range that is deleted will also be deleted.

ip nat map *list-tag priv-from-address priv-to-address pub-range-tag*

This command allows you to map a range of private addresses (from priv-from-address to priv-to-address inclusive) to the public address range named pub-range-tag. This map is appended to the list of maps named list-tag. If a map list of that name doesn't exist it is created.

---

no ip nat map [*list-tag*]

This command allows you to delete the map list named *list-tag*. This also deletes all of the maps contained in the list. No public range referred to by any of the contained maps is deleted. If a connection profile has been bound to this map list it will be updated to reflect the fact that the list no longer exists, and will thus be bound to no map list.

If the *list-tag* is omitted then all map lists will be deleted.

---

show ip nat map *list-tag*

This command allows you to see the rules contained in the nat map list named *list-tag*.

Example:

```
show ip nat map Easy-PAT
```

---

ip nat server *list-tag priv-ip-addr pub-ip-addr* { *port-name pub-start-port* [*pub-end-port*] }
{ both | tcp | udp } *priv-start-port*

This command allows you to map a private server address, *priv-ip-addr*, and port or ports to a public address *pub-ip-addr* and the same port or ports. In the Motorola Netopia® router's earlier firmware releases this feature was called "Exported Services." Its primary use is to allow servers to be accessed through a WAN interface to which PAT has been applied.

portname currently may be one of **ftp**, **telnet**, **smtp**, **tftp**, **gopher**, **finger**, **www-http**, **pop2**, **pop3**, **snmp**, **timbuktu**, **pptp**, or **irc**.

Examples:

```
ip nat server Easy-Servers 192.168.1.104 62.3.76.205 8888 8889 both 8888

ip nat server Easy-Servers 192.168.1.104 62.3.76.205 8080 8080 tcp 8080

ip nat server Easy-Servers 192.168.1.104 62.3.76.205 9000 9010 udp 19010

ip nat server Easy-Servers 192.168.1.104 62.3.76.205 smtp tcp 25
```

---

no ip nat server [*list-tag*]

This command allows you delete a list of NAT server maps. If the *list-tag* is omitted then all server lists will be deleted.

---

show ip nat server *list-tag*

This command allows you to see the rules contained in the nat server list named *list-tag*.

Example:

```
show ip nat server Easy-Servers
```

show ip nat translation

This command displays the current sessions passing through network address translation.

Example:

```
#show ip nat translations
LAN IP address--Port--WAN IP address--Port--Rem IP Address--Port--Dir-Prot-h:mm
    10.1.32.127:57037     10.1.32.127:57037  204.152.184.72:123   out UDP  0:01
   192.168.1.100:4956     10.1.32.127:49161     10.1.32.250:10    out UDP  0:02
   192.168.1.100:         10.1.32.127:          10.1.32.45:       out ICMP  8s
    10.1.32.127:23        10.1.32.127:23        10.1.32.250:1585  in  TCP  >10h
Total entries in NAT cache: 4
```

Beginning with Firmware Version 8.3.3, the session lifetime in hours:minutes format is displayed. Indefinitely long sessions greater than 10 hours, are shown as ">10h".

NAT Application Layer Gateway Commands

Note: The commands in this section are supported beginning with Firmware Version 8.2.

NAT Application Level Gateway Commands
ip nat alg [*algname*] **enable** [**yes** | **no**] **show ip nat alg** [*algname*] **enable** **no ip nat alg** [*algname*] **enable**

ip nat alg [*algname*] **enable** [**yes** | **no**]
show ip nat alg [*algname*] **enable**
no ip nat alg [*algname*] **enable**

These commands allow you to enable, disable, or show the status of the router's support for a variety of Application Layer Gateways (ALGs). An application layer gateway (ALG) is a NAT component that helps certain application sessions to pass cleanly through NAT. Each ALG has a slightly different function based on the particular application's protocol-specific requirements.

An internal client first establishes a connection with the ALG. The ALG determines if the connection should be allowed or not and then establishes a connection with the destination computer. All communications go through two connections – client to ALG and ALG to destination. The ALG monitors all traffic against its rules before deciding whether or not to forward it. The ALG is the only address seen by the public Internet so the internal network is concealed. In some situations, it may be desirable to disable some of the ALGs.

Accepted values for *algname* are:

aim	aurp	cuseeme	esp
gre	h323	ike	pptp
roadrunner	netbios-datagram		
Beginning with Firmware Version 8.3.1, the following value is also accepted:			
yahoo			
Beginning with Firmware Version 8.4, the following value is also accepted:			
sip			

Backup Configuration Commands

Backup Configuration Commands
backup enable { **no** | **manual** | **automatic** | **yes** } **no backup** **show backup status**
backup [**no**]
backup delay *1..65535*
backup ping host { *ip-address* | *host-name* }
backup recovery delay *1..65535*
backup recovery idle delay *1..65535*
backup recovery idle only { **yes** | **no** }
backup recovery layer-2-loss { **yes** | **no** }
backup recovery mode { **manual** | **automatic** }
backup failure layer-2 delay *0..65535*
no backup failure layer-2 delay
show backup failure layer-2 delay
backup recovery rip tx disable [**yes** | **no**] (supported in V8.6.1) **no backup recovery rip tx disable** **show backup recovery rip tx disable**

backup enable { **no** \| **manual** \| **automatic** \| **yes** }
no backup
show backup status

These commands allow you to set, disable, or display the status of the dial backup feature.

Note: **yes** = automatic

backup delay *1..65535*

This command allows you to set the number of seconds before the router invokes the dial backup feature in the event of loss of connectivity. This allows you to determine how long you want the system to wait before the backup port becomes enabled in the event of primary line failure, ensuring that the primary WAN connection is not merely briefly interrupted before the router switches to backup mode. Minimum is 10 seconds.

backup [**no**]

This command allows you directly to invoke or cancel the backup mode.

backup ping host { *ip-address* | *host-name* }

This command allows you to enter an IP address or resolvable DNS name that the router will ping. This is an optional item that is particularly useful for testing if the remote end of a VPN connection has gone down. Should this address become unreachable the router will treat this as a loss of connectivity and begin the backup timer. This loss is a Layer 2 loss.

backup recovery delay *1..65535*

This command allows you to set the number of seconds before the router attempts to recover back to the primary WAN connection. This allows you to determine how long you want the system to wait before the primary WAN port becomes enabled after connectivity is restored, ensuring that the backup connection is not merely briefly interrupted before the router switches to primary mode. Minimum is 10 seconds.

backup recovery idle delay *1..65535*

This command allows you to set the number of seconds for the backup link to be idle, i.e. passing no traffic, before the router attempts to recover back to the primary WAN connection.

backup recovery idle only { **yes** | **no** }

This command allows you to toggle the idle delay on or off.

backup recovery layer-2-loss { **yes** | **no** }

Note: Beginning with Firmware Version 8.3.3, this command is unncessary and consequently no longer supported.

This command allows you to specify whether the router should try to auto-recover when the backup is invoked because of a layer 2 loss, for example, no valid Connection Profile. (Layer 1 is still available, and this is what recovery checks.) Use this setting with caution. Setting it to **yes** may induce alternating switching between backup and recovery mode. This setting will determine the recovery behavior of a manual backup and ping failure backup. These two failures are treated as layer 2 failures.

backup recovery mode { **manual** | **automatic** }

This command allows you to specify the backup recovery mode to be either manually or automatically invoked.

backup failure layer-2 delay *0..65535*
no backup failure layer-2 delay
show backup failure layer-2 delay

Note: Beginning with Firmware Version 8.3.3, these commands are unncessary and consequently no longer supported.

These commands allow you to specify, disable, or show the layer-2 backup failure delay interval. Note that **l2** is an acceptable synonym for **layer-2**.

backup recovery rip tx disable [**yes** | **no**]
no backup recovery rip tx disable
show backup recovery rip tx disable

Note: These commands are supported beginning with Firmware Version 8.6.1.

These commands allow you to disable, enable, or show the status of RIP services on the primary WAN interface when the Router is in Backup mode. RIP services ordinarily will continue to run on the primary interface when in Backup mode, attempting to determine “layer 2” connectivity.

Usually, the **no** setting is desirable, since it may be required to ping the configured host that determines “layer 2” connectivity. However, RIP running on the recovering primary interface may cause problems specific to your application. When in Backup mode, RIP routes are still sent and received through the primary interface, even though that is not the active interface.

Serial port modem backup configuration commands

Serial Port Modem Backup Configuration Commands
interface serial *id* **modem directory-number** *string* **no interface serial** *id* **modem directory-number** **show interface serial** *id* **modem directory-number** *string* **interface serial** *id* **modem baud** *string* **show interface serial** *id* **modem baud** **interface serial** *id* **modem init-string** *string* **no interface serial** *id* **modem init-string** **show interface serial** *id* **modem init-string** **interface serial** *id* **mode** { **console** | **modem** } **show interface serial** *id* **mode**

interface serial *id* **modem directory-number** *string*
no interface serial *id* **modem directory-number**
show interface serial *id* **modem directory-number** *string*

These commands allow you to set, disable, or show the modem directory number for the specified serial interface. The *string* parameter can contain up to 15 characters. Non-dialable characters are allowed, but ignored.

Note: The 4000-Series data routers have one serial port, and its *id* is "0" (zero).

interface serial *id* **modem baud** *string*
show interface serial *id* **modem baud**

These commands allow you to set or show the baud rate for the specified serial port. Permissible values for the *string* parameter are 9600, 19200, 38400, 57600, 115200.

interface serial *id* **modem init-string** *string*
no interface serial *id* **modem init-string**
show interface serial *id* **modem init-string**

These commands allow you to set, delete, or show the modem initialization string for the specified serial port. Consult your modem's documentation or the AT command set for an appropriate initialization string. The *string* parameter can contain up to 31 characters.

interface serial *id* **mode** { **console** | **modem** }
show interface serial *id* **mode**

These commands allow you to set or show the mode for the specified serial port, either **console** or **modem**. In console mode, the port automatically detects the baud rate of your terminal emulation software; in modem mode, the baud rate must be specified.

RADIUS Authentication Configuration Commands

RADIUS Authentication Configuration Commands
console authentication { **local** | **radius** | **radius-local** [**serial-only**] | **local-radius** } **show console authentication** **radius-server** { **1** | **2** } { *ip-address* | *hostname* } [**secret** *secret*] **no radius-server** { **1** | **2** } **show radius-server** [**1** | **2**] **radius identifier** *identifier*

console authentication { **local** | **radius** | **radius-local** [**serial-only**] | **local-radius** }
show console authentication

These commands allow you to set or show how the router will authenticate users seeking console configuration access using a remote authentication database maintained by a RADIUS server. It supports four security database modes: **local**, **radius**, **radius-local**, and **local-radius**.

Specifying **local** selects the local router database authentication mechanism.

Specifying **radius** causes the router to ignore the local database and to authenticate users using the configured RADIUS server.

Specifying **radius-local** causes the router to attempt to authenticate a user first using a RADIUS server and then, if that fails, using the local authentication database.

Specifying **radius-local serial-only** causes the router to attempt to authenticate a user first using the configured RADIUS server(s) and then, if that fails and the user is accessing the router via the built-in serial console port, using the local authentication database. If the user is accessing the router via telnet or asynchronous dial-in (via a modem on the AUX port), and RADIUS authentication fails, the local authentication database is not consulted and the user is refused access to the router.

Specifying **local-radius** causes the router to attempt to authenticate a user first using the local authentication database, and then, if that fails using the configured RADIUS server.

In those modes that involve both RADIUS and the local database, if the local database includes no user-name/password pairs, authentication will succeed only if the RADIUS server authenticates the user. This differs from the **local** mode where no authentication is performed when the local database is empty.

radius-server { **1** | **2** } { *ip-address* | *hostname* } [**secret** *secret*]
no radius-server { **1** | **2** }
show radius-server [**1** | **2**]

These commands allow you to specify, delete, or show a RADIUS **server** either by using an IP address in dotted-quad notation or by using a hostname to be resolved using the Domain Name System (DNS) information configured in the router. In addition to specifying the server's IP address or hostname, you must also specify a shared-secret known to both the router and the RADIUS server. The **secret** is used to encrypt RADIUS transactions in transit.

radius identifier *identifier*

This command allows you to specify the RADIUS **identifier** as either an IP address in dotted-quad notation (to be used as the value of the NAS-IP-Address (4) attribute), or an arbitrary string (to be used as the value of the NAS-Identifier (32) attribute), in the router's outgoing Access-Request packets. The RADIUS **identifier** is limited to 63 characters.

TACACS+ Authentication Configuration Commands

Note: The commands in this section are supported beginning with firmware version 8.4, and supplement the RADIUS server commands in the previous section.

TACACS+ Authentication Configuration Commands
console authentication [**local** | **radius** | **radius-local** | **radius-local serial-only** | **local-radius** | **tacacs-plus** | **tacacs-plus-local** | **tacacs-plus-local serial-only** | **local-tacacs-plus**] **remote-server** { *index* } { *host* } **secret** *key* **tacacs-plus accounting** [**yes** | **no**]

console authentication
[**local** | **radius** | **radius-local** | **radius-local serial-only** | **local-radius** |
tacacs-plus | **tacacs-plus-local** | **tacacs-plus-local serial-only** | **local-tacacs-plus**]

This command sets the remote authentication protocol to RADIUS or TACACS+ and selects the ordering of the security database lookup.

remote-server { *index* } { *host* } **secret** *key*

This command sets up the primary and alternate authentication servers. It applies to both RADIUS and TACACS+. The **radius-server** command is retained for backward compatibility. If the remote authentication protocol is set to RADIUS, **show config** will display "radius-server…" rather than "remote-server…"

index: 1 = primary server; 2 = alternate server

host: IP address or DNS hostname of server

secret: RADIUS or TACACS+ shared secret

---

tacacs-plus accounting [**yes** | **no**]

This command enables or disables TACACS+ accounting.

IP Filterset Configuration Commands

Note: Beginning with firmware version 8.2, the **force-route** and **force-route-gateway** attributes have been added for policy-based routing, and the **tos** and **tos-mask** attibutes have been added for TOS field matching.

IP Filterset Configuration Commands

ip filterset *fs-tag* {**in** | **out**} [**filter**] *filter-id* [**enable** {**yes** | **no**}]
[**forward** {**yes** | **no**} | [**force-route** {**yes** | **no**] **force-route-gateway** *ip-addr*}]]
[{ **call-placement** | **idle-reset** } { **no-change** | **disabled** }]
[**source** { *ip-addr/mask-bits* | *ip-addr mask* }]
[**destination** { *ip-addr/mask-bits* | *ip-addr mask* }]
[**tos** { *tos_value* **tos-mask** *tos_mask_value* } |
[**protocol** { *1..65535* |
any |
gre |
{ {**tcp** | *6* } [{**source** *port-compare*}] [{**destination** *port-compare*}]
[**established** | **all**] } |
{ {**udp** | *17* } [{**source** *port-compare*}] [{**destination** *port-compare*}]} |
{ {**icmp** | *1* } [{**type** *port-compare*}] [{**code** *port-compare*}]} }]
no ip filterset *fs-id* [{**in** | **out**} [*filter-id*]]
show ip filterset *fs-id* [{**in** | **out**} [*filter-id*]]

The CLI for filters is fairly complex. More explanation follows the command itself.

compare-op = { **nc** | **ne** | <> | **lt** | < | **le** | <= | **eq** | = | **ge** | >= | **gt** | > }

port-compare = { **nc** | { *compare-op digits* } }

filter-id = { *1..255* | **new** | **last** }

---

ip filterset *fs-tag* {**in** | **out**} [**filter**] *filter-id* [**enable** {**yes** | **no**}]
[**forward** {**yes** | **no**} | [**force-route** {**yes** | **no**] **force-route-gateway** *ip-addr*}]
[{ **call-placement** | **idle-reset** } { **no-change** | **disabled** }]
[**source** { *ip-addr/mask-bits* | *ip-addr mask* }]
[**destination** { *ip-addr/mask-bits* | *ip-addr mask* }]
[**tos** { *tos_value* **tos-mask** *tos_mask_value* } |
[**protocol** { *1..65535* |
any |
gre |
{ {**tcp** | *6* } [{**source** *port-compare*}] [{**destination** *port-compare*}]
[**established** | **all**] } |
{ {**udp** | *17* } [{**source** *port-compare*}] [{**destination** *port-compare*}]} |
{ {**icmp** | *1* } [{**type** *port-compare*}] [{**code** *port-compare*}]} }]
no ip filterset *fs-id* [{**in** | **out**} [*filter-id*]]
show ip filterset *fs-id* [{**in** | **out**} [*filter-id*]]

set

A Filter set, as with NAT Server and Rule Lists, is instantiated by creating its first contained object. This first filter can be identified by its ones-based index, 1, or with the special **new** keyword. Subsequent modifications to this filter, assuming no more filters have been added to the filter set yet, must be done by referring to the filter either by id (1), or by the other special keyword **last**. Subsequent filters can be added using either **new** or by the next integer filter id. You can always specify the last filter in the set by using **last**. It is an error to attempt to create a new filter whose id is not 1 greater than the id of the last filter.

Using **new** and **last** allow you to create filter sets without using filter indices.

show

You can show the contents of all filter sets by typing:

```
show ip filterset
```

Or you can show the contents of a filter set by typing (for example):

```
show ip filterset "My Filters"
```

Or all of the input or output filters of a filter set by adding the {in | out} keyword:

```
show ip filterset "My Filters" in
```

Or a particular filter by specifying {in | out} and the tag:

```
show ip filterset "My Filters" in 3
```

Since the command line console is currently constrained to 78 characters per line, the show command breaks each filter up into four separate lines, for example:

```
show ip filterset "Basic Firewall" in 2
ip filterset "Basic Firewall" in 2 enable yes forward no
ip filterset "Basic Firewall" in 2 source 0.0.0.0/0
ip filterset "Basic Firewall" in 2 destination 0.0.0.0/0
```

```
ip filterset "Basic Firewall" in 2 protocol tcp source nc destination
eq 6000 any
```

Note: Some commands, when dumping existing canned filters, exceed 78 characters and will wrap. To work around this limitation use truncated keywords.

no

Syntax corresponds to the syntax for **show**.

Hardware Acceleration Configuration Commands

Hardware Acceleration Configuration Commands
hardware acceleration enable { **yes** | **no** } **no hardware acceleration enable** **show hardware acceleration enable**

hardware acceleration enable { **yes** | **no** }
no hardware acceleration enable
show hardware acceleration enable

These commands allow you to enable, disable, or show the status of hardware acceleration if the XL acceleration/encryption daughtercard is installed in the R-Series router, or if the 4000-Series router has onboard acceleration capability (XL models).

In the unlikely event of a hardware acceleration card failure, the **no hardware acceleration enable** command allows you to turn off hardware acceleration. This will disable IPcomp compression.

Global IPSec/IKE Configuration Commands

IKE Configuration Commands
ike phase1 { *name* | *index* } [{ **yes** | **no** }] **no ike phase1** { *name* | *index* } **show ike phase1** { *name* | *index* }
show ike phase1 { *name* | *index* } **id**
ike phase1 { *name* | *index* } **tag** *string* **show ike phase1** { *name* | *index* } **tag**
ike phase1 { *name* | *index* } **mode** { **main** | **aggressive** } **show ike phase1** { *name* | *index* } **mode**
ike phase1 { *name* | *index* } **identity** { **remote** | **local** } { **ipv4-address** | **ipv4-subnet** | **ipv4-range** | **hostname** | **e-mail-address** | **ascii-key-id** | **hex- key-id** } *string* **show ike phase1** { *name* | *index* } **identity** [{ **remote** | **local** }]
ike phase1 { *name* | *index* } **authentication method** { **shared-secret** } **show ike phase1** { *name* | *index* } **authentication method**
ike phase1 { *name* | *index* } **authentication shared-secret** { **ascii** | **hexadecimal** } *string*
ike phase1 { *name* | *index* } **dangling-sas** { **yes** | **no** } **show ike phase1** { *name* | *index* } **dangling-sas** **no ike phase1** { *name* | *index* } **dangling-sas**
ike phase1 { *name* | *index* } **encryption** { **des** | **3des** } **show ike phase1** { *name* | *index* } **encryption**
ike phase1 { *name* | *index* } **group** { **1** | **2** | **5** | **dh-768-bits** | **dh-1024-bits** | **dh-1536-bits** } **show ike phase1** { *name* | *index* } **group**
ike phase1 { *name* | *index* } **hash** { **sha1** | **md5** } **show ike phase1** { *name* | *index* } **hash**
show ike status
show ipsec sessions

Global IPSec Configuration Commands
ike phase1 { *name* | *index* } **independent rekeys** { **yes** | **no** } **show ike phase1** { *name* | *index* } **independent rekeys** **no ike phase1** { *name* | *index* } **independent rekeys** **ike phase1** { *name* | *index* } **initial-contact** { **yes** | **no** } **show ike phase1** { *name* | *index* } **initial-contact** **no ike phase1** { *name* | *index* } **initial-contact** **ike phase1** { *name* | *index* } **negotiation** { **normal** | **initiate-only** | **respond-only** } **show ike phase1** { *name* | *index* } **negotiation** **ike phase1** { *name* | *index* } **pfs** { **yes** | **no** } **show ike phase1** { *name* | *index* } **pfs** **no ike phase1** { *name* | *index* } **pfs** **ike phase1** { *name* | *index* } **port policy** { **"strict"** | **"permissive"** } **show ike phase1** { *name* | *index* } **port policy** **ike phase1** { *name* | *index* } **sa lifetime** { **seconds** | **kbytes** } { *non-negative-integer* | **none** } **show ike phase1** { *name* | *index* } **sa lifetime** [{ **seconds** | **kbytes** }] **no ike phase1** { *name* | *index* } **sa lifetime** [{ **seconds** | **kbytes** }] **ike phase1** { *name* | *index* } **sa use-policy** { **new-sas-immediately** | **old-sas-until-expired** } **show ike phase1** { *name* | *index* } **sa use-policy** **ike phase1** { *name* | *index* } **vendor-id** { **yes** | **no** } **show ike phase1** { *name* | *index* } **vendor-id** **no ike phase1** { *name* | *index* } **vendor-id**

---

ike phase1 { *name* | *index* } [{ **yes** | **no** }]
no ike phase1 { *name* | *index* }
show ike phase1 { *name* | *index* }

These commands create, delete, or show the specified IKE Phase1 profile, which may be identified by index or by name. The show version of this command displays the value **yes** if the specified IKE Phase 1 Profile exists, and **no** otherwise.

---

show ike phase1 { *name* | *index* } **id**

This command displays the index of the specified IKE Phase1 profile. This command is useful only when referring to a profile by name.

ike phase1 { *name* | *index* } **tag** *string*
show ike phase1 { *name* | *index* } **tag**

These commands name or display the specified IKE Phase1 profile.

ike phase1 { *name* | *index* } **mode** { **main** | **aggressive** }
show ike phase1 { *name* | *index* } **mode**

These commands set or display whether the specified IKE Phase1 profile uses main mode or aggressive mode.

ike phase1 { *name* | *index* } **identity** { **remote** | **local** } { **ipv4-address** | **ipv4-subnet** | **ipv4-range** | **hostname** | **e-mail-address** | **ascii-key-id** | **hex- key-id** } *string*
show ike phase1 { *name* | *index* } **identity** [{ **remote** | **local** }]

These commands set or display the specified IKE Phase1 profile's local or remote identity type and value.

The **identity** type specifies the type of Identity value to be used. Possible types are: **ipv4-address**, **ipv4-subnet**, **ipv4-range**, **hostname**, **e-mail-address**, **ascii-key-id**, and **hex-key-id**. The identity value specifies a value of the specified type as follows:

Identity Type	Format of Identity Value
IPv4 Address	A single IPv4 address in the familiar dotted-quad notation (a.b.c.d)
IPv4 Subnet	A single IPv4 network address in the familiar dotted-quad notation (a.b.c.d) followed by a mask specified EITHER by a slash and a bit-count between 0 and 32 OR by a second dotted-quad.
IPv4 Range	Two IPv4 addresses in the familiar dotted quad notation (a.b.c.d) separated by a space.
Host Name	A fully-qualified domain name (FQDN)
E-Mail Address	An RFC 822 e-mail address in the form user@hostname
Key ID (ASCII)	An opaque string consisting of printable ASCII characters represented as a sequence of printable ASCII characters
Key ID (HEX)	An opaque string consisting of arbitrary 8-bit ASCII values represented as a sequence of HEXADECIMAL digits, each of which corresponds to one nibble of the string value

ike phase1 { *name* | *index* } **authentication method** { **shared-secret** }
show ike phase1 { *name* | *index* } **authentication method**

These commands set or display the specified IKE Phase1 profile's authentication method. Currently, the only supported method is shared-secret; others may be added in the future.

---

ike phase1 { *name* | *index* } **authentication shared-secret** { **ascii** | **hexadecimal** } *string*

This command sets the specified IKE Phase1 profile's shared secret. For security reasons no **show** variant of this command exists.

---

ike phase1 { *name* | *index* } **dangling-sas** { **yes** | **no** }
show ike phase1 { *name* | *index* } **dangling-sas**
no ike phase1 { *name* | *index* } **dangling-sas**

These commands set, display, or disable whether or not Phase 2 SAs may persist after the underlying Phase 1 SAs have expired.

---

ike phase1 { *name* | *index* } **encryption** { **des** | **3des** }
show ike phase1 { *name* | *index* } **encryption**

These commands set or display the specified IKE Phase1 profile's encryption algorithm.

---

ike phase1 { *name* | *index* } **group** { **1** | **2** | **5** | **dh-768-bits** | **dh-1024-bits** | **dh-1536-bits** }
show ike phase1 { *name* | *index* } **group**

These commands set or display the specified IKE Phase1 profile's Diffie-Hellman group.

Note: **1** and **dh-768-bits**, **2** and **dh-1024-bits**, and **5** and **dh-1536-bits**, respectively, are synonyms.

---

ike phase1 { *name* | *index* } **hash** { **sha1** | **md5** }
show ike phase1 { *name* | *index* } **hash**

These commands set or display the specified IKE Phase1 profile's hash algorithm.

---

show ike status

Note: This command is supported beginning with firmware version 8.5.

This command allows you to view an IKE session setup.

---

show ipsec sessions

Note: This command is supported beginning with firmware version 8.5.

This command allows you to view an IPsec session setup.

---

ike phase1 { *name* | *index* } **independent rekeys** { **yes** | **no** }
show ike phase1 { *name* | *index* } **independent rekeys**
no ike phase1 { *name* | *index* } **independent rekeys**

These commands set or display the specified IKE Phase1 profile's independent phase 2 re-keys setting.

---

ike phase1 { *name* | *index* } **initial-contact** { **yes** | **no** }
show ike phase1 { *name* | *index* } **initial-contact**
no ike phase1 { *name* | *index* } **initial-contact**

These commands set or display the specified IKE Phase1 profile's send initial-contact message setting.

---

ike phase1 { *name* | *index* } **negotiation** { **normal** | **initiate-only** | **respond-only** }
show ike phase1 { *name* | *index* } **negotiation**

These commands set or display the specified IKE Phase1 profile's negotiation setting.

---

ike phase1 { *name* | *index* } **pfs** { **yes** | **no** }
show ike phase1 { *name* | *index* } **pfs**
no ike phase1 { *name* | *index* } **pfs**

These commands set, display, or disable the specified IKE Phase1 profile's perfect forward secrecy setting.

---

ike phase1 { *name* | *index* } **port policy** { **"strict"** | **"permissive"** }
show ike phase1 { *name* | *index* } **port policy**

These commands set or display whether or not IKE requires packets to originate from the IANA port (500).

---

ike phase1 { *name* | *index* } **sa lifetime** { **seconds** | **kbytes** } { *non-negative-integer* | **none** }
show ike phase1 { *name* | *index* } **sa lifetime** [{ **seconds** | **kbytes** }]
no ike phase1 { *name* | *index* } **sa lifetime** [{ **seconds** | **kbytes** }]

These commands set, display, or disable one or both of the specified IKE Phase1 profile's two SA lifetimes (in seconds and/or kilobytes protected). Specifying neither the keyword **seconds** nor the keyword **kbytes** with the show variant of this command displays both lifetime values. The keyword **none** is equivalent to the value zero, and indicates that there is no lifetime of the specified type. The Phase1 SA lifetime minimum is 300 (seconds) and the maximum is 1 (leap) year (31622400 seconds).

Note: It is a run-time checked error if both of the IKE Phase 1 profile's SA lifetime values are set to zero or **none**.

---

ike phase1 { *name* | *index* } **sa use-policy** { **new-sas-immediately** | **old-sas-until-expired** }
show ike phase1 { *name* | *index* } **sa use-policy**

These commands set or display the specified IKE Phase1 profile's SA use policy.

---

ike phase1 { *name* | *index* } **vendor-id** { **yes** | **no** }
show ike phase1 { *name* | *index* } **vendor-id**
no ike phase1 { *name* | *index* } **vendor-id**

These commands set, display, or disable the specified IKE Phase1 profile's send vendor-id payload setting.

IKE Dead Peer Detection

Note: The commands in this section are supported beginning with Firmware Version 8.5.2.

IKE Dead Peer Detection Configuration Commands
ike phase1 { *name* | *index* } **dead-peer-detection enable** { **yes** | **no** } **show ike phase1** { *name* | *index* } **dead-peer-detection enable** **no ike phase1** { *name* | *index* } **dead-peer-detection enable** **show ike phase1** { *name* | *index* } **dead-peer-detection timeout** **ike phase1** { *name* | *index* } **dead-peer-detection timeout** { **3-65535** }

ike phase1 { *name* \| *index* } **dead-peer-detection enable** { **yes** \| **no** }
show ike phase1 { *name* \| *index* } **dead-peer-detection enable**
no ike phase1 { *name* \| *index* } **dead-peer-detection enable**

These commands allow you to enable, disable, or show the status of the traffic-based IKE dead peer detection feature. Traffic-based IKE dead peer detection allows IKE to negotiate RFC3706-based IKE “keepalives” with a remote security gateway (IKE peer) that supports them. Default is **no** (disabled).

show ike phase1 { *name* \| *index* } **dead-peer-detection timeout**
ike phase1 { *name* \| *index* } **dead-peer-detection timeout** { **3-65535** }

These commands allow you to specify or show an interval, from 3 to 65535 seconds, during which IPSec traffic may be idle before the router sends a keepalive message to its peer, when **ike phase1 dead-peer-detection enable** is set to **yes**. Default is 20 (seconds).

Xauth configuration commands

Note: The commands in this section are supported beginning with Firmware Version 8.4.2.

Xauth Configuration Commands
ike phase1 { *name* | *index* } **xauth mode** { **disabled** | **client** | **concentrator** } **show ike phase1** { *name* | *index* } **xauth mode** **ike phase1** { *name* | *index* } **xauth database** { **local** | **radius** } **show ike phase1** { *name* | *index* } **xauth database** **ike phase1** { *name* | *index* } **xauth username** *name* **show ike phase1** { *name* | *index* } **xauth username** **ike phase1** { *name* | *index* } **xauth password** *password* **show ike phase1** { *name* | *index* } **xauth password**

ike phase1 { *name* | *index* } **xauth mode** { **disabled** | **client** | **concentrator** }
show ike phase1 { *name* | *index* } **xauth mode**

These commands allow you to enable, disable, or show the status of the Xauth extensions to IPsec, as **client** or **concentrator** when **ike phase1 mode** is set to **aggressive**. Default is **disabled**.

ike phase1 { *name* | *index* } **xauth database** { **local** | **radius** }
show ike phase1 { *name* | *index* } **xauth database**

These commands allow you to specify the authentication database type to be used for Xauth user authentication when **xauth mode** is set to **concentrator**. If you specify **radius**, the Router will use the globally-configured RADIUS server database. (IF **xauth mode** is set to **client**, it can only send the locally-configured username/password.) See "RADIUS Authentication Configuration Commands" on page 2-119.

ike phase1 { *name* | *index* } **xauth username** *name*
show ike phase1 { *name* | *index* } **xauth username**

These commands allow you to set or show the the Xauth username specified by *name*.

ike phase1 { *name* | *index* } **xauth password** *password*
show ike phase1 { *name* | *index* } **xauth password**

These commands allow you to set or show the the Xauth password specified by *password*.

Current Restrictions

None.

Chapter 3

Motorola Netopia® Router Connection Profile Commands

This chapter describes the syntax of the supported command set for creating and modifying Connection Profiles on the Motorola Netopia® R-series, 4000-series, and 3000 Enterprise-series Router families.

Connection Profile Commands

Connection Profile Commands
cp { *name* | *index* } **no cp** { *name* | *index* } **show cp** { *name* | *index* }
cp { *name* | *index* } **enable** { **yes** | **no** } **no cp** { *name* | *index* } **enable** **show cp** { *name* | *index* } **enable**
cp { *name* | *index* } **tag** *string* **show cp** { *name* | *index* } **tag**
cp { *name / index* } **dle** { **hdlc** | **ppp** | **frame-relay** | **rfc1483** | **atmp** | **pptp** | **ipsec** | **l2tp** } **show cp** { *name* | *index* } **dle**
cp { *name / index* } **pppoe pppoa-autodetect** [**yes** | **no**] **show cp** { *name / index* } **pppoe pppoa-autodetect** **Note:** The two commands above are supported beginning with firmware release 8.5.
cp { *name* | *index* } **filterset** *string* **no cp** { *name* | *index* } **filterset** [*string*] **show cp** { *name* | *index* } **filterset**
cp { *name* | *index* } **ip addressing** { **numbered** | **unnumbered** } **show cp** { *name* | *index* } **ip addressing**
cp { *name* | *index* } **ip address local** { *ip-addr* | *ip-addr/mask-bits* | *ip-addr mask* } **no cp** { *name* | *index* } **ip address local** **show cp** { *name* | *index* } **ip address local**
cp { *name* | *index* } **ip address remote** { *ip-addr* | *ip-addr/mask-bits* | *ip-addr mask* } **no cp** { *name* | *index* } **ip address remote** **show cp** { *name* | *index* } **ip address remote**
cp { *name* | *index* } **ip dhcp client mode** { **standard** | **copper-mountain** | **cmn** } **show cp** { *name* | *index* } **ip dhcp client mode**

Connection Profile Commands (continued, 1)

show cp { *name* | *index* } **ip dhcp client status**
Note: The command above is supported beginning with firmware release 8.5.

show cp { *name* | *index* } **ip dhcp client** [**renew** | **release**]
Note: The command above is supported beginning with firmware release 8.5.

cp { *name* | *index* } **ip mask local** *ip-mask*
no cp { *name* | *index* } **ip mask local**
show cp { *name* | *index* } **ip mask local**

cp { *name* | *index* } **ip mask remote** *ip-mask*
no cp { *name* | *index* } **ip mask remote**
show cp { *name* | *index* } **ip mask remote**

cp { *name* | *index* } **ip multicast-fwd** { **yes** | **no** }
no cp { *name* | *index* } **ip multicast-fwd**
show cp { *name* | *index* } **ip multicast-fwd**

cp { *name* | *index* } **ip negotiate-lan** { **yes** | **no** }
no cp { *name* | *index* } **ip negotiate-lan**
show cp { *name* | *index* } **ip negotiate-lan**

cp { *name* | *index* } **ip netbios proxy enable** { **yes** | **no** }
no cp { *name* | *index* } **ip netbios proxy enable**
show cp { *name* | *index* } **ip netbios proxy enable**

cp { *name* | *index* } **ip rip receive** { **no** | **v1** | **v2** | **both** | **v2-md5** }
no cp { *name* | *index* } **ip rip receive**
show cp { *name* | *index* } **ip rip receive**

cp { *name* | *index* } **ip rip exclude-wan-routes**
no cp { *name* | *index* } **ip rip exclude-wan-routes**
show cp { *name* | *index* } **ip rip exclude-wan-routes**

cp { *name* | *index* } **ip rip transmit** { **no** | **v1** | **v2broadcast** | **v2multicast** | **v2broadcast-md5** | **v2multicast-md5** }
no cp { *name* | *index* } **ip rip transmit**
show cp { *name* | *index* } **ip rip transmit**

Connection Profile Commands (continued, 2)

cp { *name* | *index* } **ppp authentication type** { **none** | **pap** | **chap** }
no cp { *name* | *index* } **ppp authentication type**
show cp { *name* | *index* } **ppp authentication type**

cp { *name* | *index* } **ppp authentication** { **send** | **receive** } **name** *string*
no cp { *name* | *index* } **ppp authentication** { **send** | **receive** } **name**
show cp { *name* | *index* } **ppp authentication** { **send** | **receive** } **name**

cp { *name* | *index* } **ppp authentication** { **send** | **receive** } **password** *string*
no cp { *name* | *index* } **ppp authentication** { **send** | **receive** } **password**

cp { *name* | *index* } **ppp usage** { **1** | **2** [**preemptible**] [**dynamic**] }
show cp { *name* | *index* } **ppp usage**

cp { *name* | *index* } **frame-relay dlci auto-detect** { **yes** | **no** }
no cp { *name* | *index* } **frame-relay dlci auto-detect**
show cp { *name* | *index* } **frame-relay dlci auto-detect**

cp { *name* | *index* } **frame-relay dlci multicast-number** { 0 | 16 ... 991 }
no cp { *name* | *index* } **frame-relay multicast-number**
show cp { *name* | *index* } **frame-relay dlci multicast-number**

cp { *name* | *index* } **telco direction** { **in** | **out** | **both** }
show cp { *name* | *index* } **telco direction**

Connection Profile Commands (continued, 3)

cp { *name* | *index* } **telco dn** [**1** | **2**] *string*
no cp { *name* | *index* } **telco dn** [**1** | **2**]
show cp { *name* | *index* } **telco dn** [**1** | **2**]

cp { *name* | *index* } **telco prefix** *string*
no cp { *name* | *index* } **telco prefix**
show cp { *name* | *index* } **telco prefix**

cp { *name* | *index* } **telco callback** { **yes** | **no** }
no cp { *name* | *index* } **telco callback**
show cp { *name* | *index* } **telco callback**

cp { *name* | *index* } **ip nat enable** { **yes** | **no** }
no cp { *name* | *index* } **ip nat enable**
show cp { *name* | *index* } **ip nat enable**

cp { *name* | *index* } **ip nat map-list** *list-tag*
no cp { *name* | *index* } **ip nat map-list**
show cp { *name* | *index* } **ip nat map-list**

cp { *name* | *index* } **ip nat server-list** *list-tag*
no cp { *name* | *index* } **ip nat server-list**
show cp { *name* | *index* } **ip nat server-list**

show cp { *name* | *index* } **id**

cp { *name* | *index* } **connection demand** { **yes** | **no** }
cp { *name* | *index* } **connection timeout** *seconds*

Connection Profile Commands (continued, 4)
Connection Profile PPTP Commands
cp { *name* | *index* } **pptp ip partner** *ip-addr* **cp** { *name* | *index* } **pptp ip via** *ip-addr* **cp** { *name* | *index* } **pptp authentication type** { **pap** | **chap** | **mschap** } **cp** { *name* | *index* } **pptp compression** { **none** | **standardlzs** } **no cp** { *name* | *index* } **pptp compression** **cp** { *name* | *index* } **pptp encryption** { **none** | **mppe** } **no cp** { *name* | *index* } **pptp encryption** **cp** { *name* | *index* } **pptp authentication** { **send** | **receive** } **name** *string* **no cp** { *name* | *index* } **pptp authentication** { **send** | **receive** } **name** **show cp** { *name* | *index* } **pptp authentication** { **send** | **receive** } **name** **cp** { *name* | *index* } **pptp authentication** { **send** | **receive** } **password** *string* **no cp** { *name* | *index* } **pptp authentication** { **send** | **receive** } **password**
Connection Profile Manual Connect/Disconnect Commands
connect cp { *name* | *index* } **disconnect cp** { *name* | *index* }
Connection Profile Backup Configuration Commands
cp { *name* / *index* } **interface-group** { **primary** | **backup** | **auxiliary** } **show cp** { *name* / *index* } **interface-group**

Connection Profile Commands (continued, 5)

Connection Profile RIP-2 MD5 Configuration Commands

cp *id* **ip rip auth key** *id*
no cp *id* **ip rip auth key** *id*
show config cp *id* **ip rip auth key**

cp *id* **ip rip auth key** *id* **start date** *date*
show cp *id* **ip rip auth key** *id* **start date**

cp *id* **ip rip auth key** *id* **start time** *time*
show cp *id* **ip rip auth key** *id* **start time**

cp *id* **ip rip auth key** *id* **end date** *date*
show cp *id* **ip rip auth key** *id* **end date**

cp *id* **ip rip auth key** *id* **end time** *time*
show cp *id* **ip rip auth key** *id* **end time**

cp *id* **rip auth key** *id* **end time mode** { **infinite** | **date** }
show cp *id* **ip rip auth key** *id* **end time mode**

cp *id* **rip auth key** *id* **key** <*string*>

Connection Profile IP NAT Passthrough Commands

cp { *name* | *index* } **ip nat passthrough enable** { **yes** | **no** }
no cp { *name* | *index* } **ip nat passthrough enable**
show cp { *name* | *index* } **ip nat passthrough enable**

cp { *name* | *index* } **ip nat passthrough dhcp enable** { **yes** | **no** }
no cp { *name* | *index* } **ip nat passthrough dhcp enable**
show cp { *name* | *index* } **ip nat passthrough dhcp enable**

cp { *name* | *index* } **ip nat passthrough dhcp mac-address** { *mac-address* }
show cp { *name* | *index* } **ip nat passthrough dhcp mac-address**

Connection Profile Commands (continued, 6)
Connection Profile Stateful Inspection Commands
cp { *name* | *index* } **ip state-insp enable** { **yes** | **no** | **on** | **off** } **no cp** { *name* | *index* } **ip state-insp enable** **show cp** { *name* | *index* } **ip state-insp enable** **cp** { *name* | *index* } **ip state-insp xposed-list** *xposed-list_name* **no cp** { *name* | *index* } **ip state-insp xposed-list** **show cp** { *name* | *index* } **ip state-insp xposed-list** **cp** { *name* | *index* } **ip state-insp tcp-seq-diff** *diff* **show cp** { *name* | *index* } **ip state-insp tcp-seq-diff** **cp** { *name* | *index* } **ip state-insp router-access** { **yes** | **no** | **on** | **off** } **no cp** { *name* | *index* } **ip state-insp router-access** **show cp** { *name* | *index* } **ip state-insp router-access** **cp** { *name* | *index* } **ip state-insp deny-frag** { **yes** | **no** | **on** | **off** } **no cp** { *name* | *index* } **ip state-insp deny-frag** **show cp** { *name* | *index* } **ip state-insp deny-frag**

Connection Profile Commands (continued, 7)

Connection Profile L2TP Configuration Commands

cp { *name* | *index* } **l2tp ip partner** *ip-addr*
show cp { *name* | *index* } **l2tp ip partner**

cp { *name* | *index* } **l2tp ip via** *ip-addr*
show cp { *name* | *index* } **l2tp ip via**

cp { *name* | *index* } **l2tp authentication enable** { **yes** | **no**}
show cp { *name* | *index* } **l2tp authentication enable**

cp { *name* | *index* } **l2tp authentication passphrase** *string*

cp { *name* | *index* } **l2tp authentication ppp type** { **pap** | **chap** }
show cp { *name* | *index* } **l2tp authentication ppp**

cp { *name* | *index* } **l2tp authentication ppp** { **send** | **receive** } **name** *string*
show cp { *name* | *index* } **l2tp authentication ppp** { **send** | **receive** } **name**

cp { *name* | *index* } **l2tp authentication ppp** { **send** | **receive** } **password** *string*
show cp { *name* | *index* } **l2tp authentication ppp** { **send** | **receive** } **password**

cp { *name* | *index* } **l2tp compression** { **none** | **standardlzs** }
no cp { *name* | *index* } **l2tp compression**
show cp { *name* | *index* } **l2tp compression**

Connection Profile GRE Tunnel Configuration Commands

cp { *name* | *index* } **gre ip partner** *ip-addr*
show cp { *name* | *index* } **gre ip partner**

cp { *name* | *index* } **gre ip via** *ip-addr*
show cp { *name* | *index* } **gre ip via**

cp { *name* | *index* } **gre checksum** [**yes** | **no**]
show cp { *name* | *index* } **gre checksum** [**yes** | **no**]

cp { *name* | *index* } **gre sequence-datagrams** [**yes** | **no**]
show cp { *name* | *index* } **gre sequence-datagrams** [**yes** | **no**]

cp { *name* | *index* } **gre key** [0..232-1]
show cp { *name* | *index* } **gre key** [0..232-1]

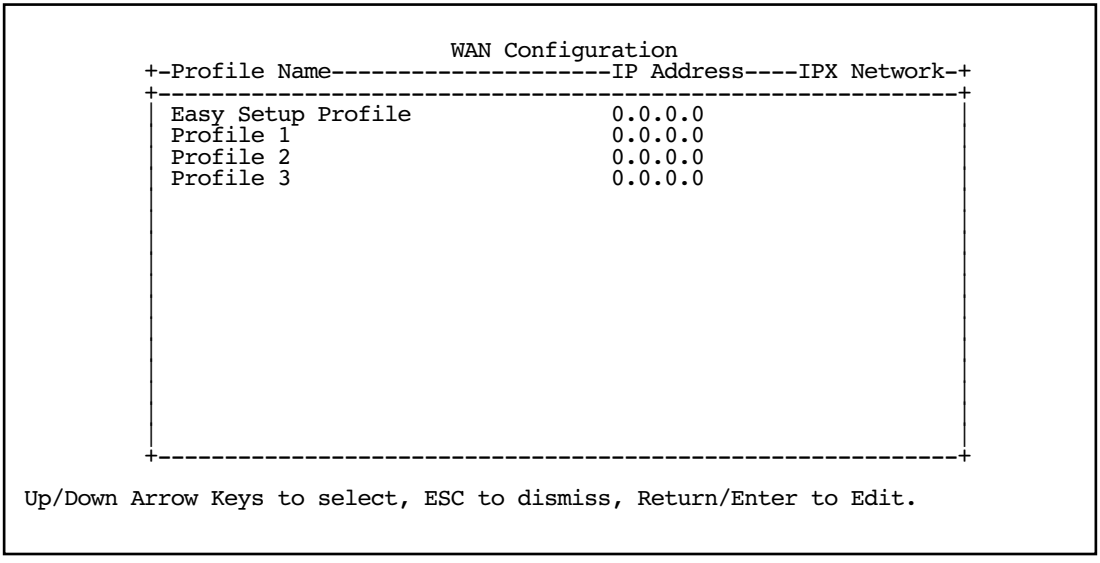
Note on Connection Profile numbering sequence

The Easy Setup Profile is always assumed to be the Primary Connection Profile, whether or not it exists. The menu console reserves the index number 1 (one) for the Easy Setup Profile, even if you do not create an Easy Setup Profile.

If you do not create an Easy Setup Profile using the Easy Setup screens, but instead use the WAN Configuration/Add Connection Profile screen to create a Connection Profile, the menu console will name it *Profile 1* by default (you can rename it anything you want). Nevertheless, the router will always assign such a profile the index number 2 (two). Profiles added subsequently are internally indexed incrementally.

This can be confusing, when issuing CLI commands because it is possible for *Profile 1* to be indexed by the router as the second profile. *Profile 2* is indexed as the third, and so on.

This is illustrated in the following menu console screen:



---

cp { *name* | *index* }

This command allows you to create a connection profile, which is a structure used to define WAN connections. *index* can be any value from 1 to 16. The command has no effect if the profile already exists. The name of the profile defaults to "profile *index*" (e.g., "profile 1").

---

no cp { *name* | *index* }

This command allows you to delete a connection profile.

---

cp { *name* | *index* } **enable** { **yes** | **no** }

This command allows you to enable or disable connection profile index. Use this command if you want to temporarily disable a profile but do not want to lose its configuration.

---

cp { *name* | *index* } **tag** *string*

This command allows you to name connection profile string, which can be up to 32 characters long.

---

cp { *name* / *index* } **dle** { **hdlc** | **ppp** | **frame-relay** | **rfc1483** | **atmp** | **pptp** | **ipsec** | **l2tp** }

This command allows you to set the encapsulation type that the connection profile will use when establishing a WAN connection. Note that a profile whose encapsulation type is incompatible with the global encapsulation type is essentially disabled and cannot be used. Also, when a profile is created, it inherits the global encapsulation type by default and thus it is not usually necessary to explicitly set this parameter.

Note: **atmfuni** is accepted as a synonym for **rfc1483**, and **frame-relay** is accepted as a synonym for **rfc1490**. **l2tp** is supported beginning with Firmware Version 8.2.
Beginning with Firmware Version 8.3.1, the following optional modifiers are permitted if the specified DLE is **ppp**:

pppoe-enable [{ **yes** | **no** }] { **llcsnap** | **vcmux** | **vc-multiplexed** }

and the following optional modifiers are permitted if the specified DLE is **rfc1483**:

{ **bridged** | **routed** }

---

cp { *name* / *index* } **pppoe pppoa-autodetect** [**yes** | **no**]
show cp { *name* / *index* } **pppoe pppoa-autodetect**

Note: These commands are supported beginning with firmware version 8.5.

These commands allow you to enable, disable, or show the status of PPPoE/PPPoA autodetection.

If you are using PPP, and you have selected **pppoe-enable**, you can further enable the ability to connect automatically to your ISP's central office equipment whether they are using PPP over Ethernet or PPP over ATM. If your ISP is using PPPoE, the connection will be made normally. If your ISP is using PPPoA, when the Motorola Netopia® Gateway detects this, it will automatically switch to PPPoA transparently.

---

cp { *name* | *index* } **filterset** *string*
no cp { *name* | *index* } **filterset** [*string*]
show cp { *name* | *index* } **filterset**

These commands allow you to associate a filter set with the a connection profile. The filter set is identified by name.

---

cp { *name* | *index* } **ip addressing** { **numbered** | **unnumbered** }

This command allows you to specify whether or not the WAN interface using this profile has an IP address. With point-to-point connections, a WAN IP address is not necessary for the router to function properly, but may be required, depending on how the equipment at the other end is configured.

---

cp { *name* | *index* } **ip address local** { *ip-addr* | *ip-addr/mask-bits* | *ip-addr mask* }

This command allows you to set the profile's local WAN IP address.

---

cp { *name* | *index* } **ip address remote** { *ip-addr* | *ip-addr/mask-bits* | *ip-addr mask* }

This command allows you to set the profile's remote WAN IP address.

---

cp { *name* | *index* } **ip dhcp client mode** { **standard** | **copper-mountain** | **cmn** }
show cp { *name* | *index* } **ip dhcp client mode**

These commands allow you to set or show the router's DHCP mode, whether **standard**, **copper-mountain**, or **cmn**.

The connection profile, default profile, and IP configuration structures now include a **dhcp client mode** setting that selects between the **standard** RFC 2131 standards-based mode of operation (the default), and the **copper-mountain** or **cmn** proprietary mode of operation.

When the DHCP client is activated on a RFC1483 MER interface, it examines the **dhcp client mode** in the associated connection profile (or the default profile there was no explicitly configured connection profile). If the **dhcp client mode** specifies **standard**, the DHCP client initializes the htype and hlen fields in the header of its DHCP requests to the appropriate values for an RFC1483 MER interface (htype = 1 and hlen = 6). If the **dhcp client mode** specifies **copper-mountain** or **cmn**, the DHCP client initializes the htype and hlen fields in the header of its DHCP requests to zero.

When the DHCP client is activated on an Ethernet WAN interface, it examines the **dhcp client mode** in the associated IP configuration structure, and behaves as described above for the RFC1483 MER DHCP client.

Note: **cmn** is accepted as a synonym for **copper-mountain**.

---

show cp { *name* | *index* } **ip dhcp client status**

Note: The command above is supported beginning with firmware version 8.5.

This command allows you to display the status of the ethernet WAN being served via DHCP for the specified connection profile. It displays:

IP Address	IP Subnet Mask

IP Gateway	DHCP server
DNS server 1	DNS server 2 (if any)
Lease Expiration	

---

show cp { *name* | *index* } **ip dhcp client** [**renew** | **release**]

Note: This command is supported beginning with firmware version 8.5.

This command allows you to renew or release the ethernet WAN IP address lease being served via DHCP for the specified connection profile.

---

cp { *name* | *index* } **ip mask local** *ip-mask*

This command allows you to set the profile's local WAN IP mask.

---

cp { *name* | *index* } **ip mask remote** *ip-mask*

This command allows you to set the profile's remote WAN IP mask.

---

cp { *name* | *index* } **ip multicast-fwd** { **yes** | **no** }
no cp { *name* | *index* } **ip multicast-fwd**
show cp { *name* | *index* } **ip multicast-fwd**

These commands allow you to set, show, or disable multicast forwarding on the specified connection profile.

---

cp { *name* | *index* } **ip negotiate-lan** { **yes** | **no** }
no cp { *name* | *index* } **ip negotiate-lan**
show cp { *name* | *index* } **ip negotiate-lan**

These commands allow you to set, delete, or show whether the specified connection profile will attempt to negotiate the router hub's IP address and subnet mask from the central site router.

The firmware includes PPP support for the IPCP Subnet Mask option documented in *PPP Internet Protocol Control Protocol Extensions for IP Subnet*, draft-helenius-ppp-subnet-00.txt. This option, together with the IPCP IP Address option, allows a central site router to supply an entire IP subnet, rather than a single IP address, for use by a CPE router.

PPP Ethernet LAN reconfiguration is controlled by an **ip negotiate-lan** connection profile flag. If the applicable connection profile specifies an unnumbered, non-NAT connection and the **ip negotiate-lan** flag is **yes**, PPP will attempt to negotiate both an IP Address and subnet mask.

cp { *name* | *index* } **ip netbios proxy enable** { **yes** | **no** }
no cp { *name* | *index* } **ip netbios proxy enable**
show cp { *name* | *index* } **ip netbios proxy enable**

These commands allow you to enable, disable, or show the NetBIOS proxy status for the specified Connection Profile. The NetBIOS proxy enables the ability to forward Windows Networking NetBIOS broadcasts. This is useful for, for example, a Virtual Private Network, in which you want to be able to browse the remote network to which you are tunnelling, as part of your Windows Network Neighborhood.

Routed connections, such as VPNs, can not use NetBEUI to carry the Network Neighborhood information. They need to use NetBIOS, because NetBEUI cannot be routed. This feature will allow browsing the Network Neighborhood without any additional workstation configuration.

Note: Microsoft Network browsing is available with or without a Windows Internet Name Service (WINS) server. Shared volumes on the remote network are accessible with or without a WINS server. Local LAN shared volumes that have Port Address Translation (PAT) applied to them are *not* available to hosts on the remote LAN. For tunnelled traffic, NAT on the WAN has no effect on the Microsoft Networking traffic.

cp { *name* | *index* } **ip rip receive** { **no** | **v1** | **v2** | **both** | **v2-md5** }

This command allows you to set the RIP receive behavior when the profile is used for a WAN connection.

cp { *name* | *index* } **ip rip exclude-wan-routes**
no cp { *name* | *index* } **ip rip exclude-wan-routes**
show cp { *name* | *index* } **ip rip exclude-wan-routes**

Note: These commands are supported beginning with Firmware Version 8.7.

These commands allow you to specify, disable, or show the status of broadcasting WAN routes via RIP. This is available only if **rip transmit** is enabled for the connection profile. The default is **no**, but if enabled, will drop any RIP routes with non-LANside information from RIP updates sent over the interface.

cp { *name* | *index* } **ip rip transmit** { **no** | **v1** | **v2broadcast** | **v2multicast** | **v2broadcast-md5** | **v2multicast-md5** }

This command allows you to set the RIP transmit behavior when the profile is used for a WAN connection.

Note: If network address translation is enabled, RIP transmit is disabled regardless of the current setting of this parameter.

cp { *name* | *index* } **ppp authentication type** { **none** | **pap** | **chap** }

This command allows you to configure the type of authentication used by the profile.

cp { *name* | *index* } **ppp authentication** { **send** | **receive** } **name** *string*

This command allows you to configure the send or receive PPP authentication name. The send name is used when the remote side attempts to authenticate the Motorola Netopia® router, and the receive name is used when the Motorola Netopia® router is attempting the authentication (for instance, if a WAN connection is being established to the router).

---

cp { *name* | *index* } **ppp authentication** { **send** | **receive** } **password** *string*

This command allows you to configure the send or receive PPP authentication password (or secret) associated with the send or receive names.

---

cp { *name* | *index* } **ppp usage** { **1** | **2** [**preemptible**] [**dynamic**] }

This command allows you to configure the characteristics of how the channels of the interface are used. The number indicates the maximum number of channels to use.

If you specify the keyword **preemptible** and more than one channel is being used for the connection, additional calls (both data and voice, when applicable) may borrow a channel for their own use.

If you specify the keyword **dynamic** channels are added and removed from the connection based on bandwidth usage. If traffic exceeds a certain threshold for a certain amount of time, and if there is a free channel available, it will be used for the connection. Conversely, if more than one channel is being used by the connection and traffic drops below a certain level for a certain amount of time, a channel will be dropped.

The keywords **dynamic** and **preemptible** may be specified only if the number of channels is 2.

Note: With the current firmware, a dynamic 2B Channel profile will also be preemptible, regardless of whether or not the **preemptible** keyword is specified.

Examples:

These examples illustrate all forms of the command that you are likely to use:

```
cp 1 ppp usage 1

cp 1 ppp usage 2

cp 1 ppp usage 2 preemptible

cp 1 ppp usage 2 dynamic
```

---

cp { *name* | *index* } **frame-relay dlci auto-detect** { **yes** | **no** }

This command allows you to enable or disable the automatic detection of Frame Relay DLCIs when the profile is used to establish a WAN connection.

---

cp { *name* | *index* } **frame relay dlci multicast-number** { 0 | 16 ... 991 }

This command allows you to specify the DLCI multicast number for the profile.

---

cp { *name* | *index* } **telco direction** { **in** | **out** | **both** }

This command allows you to set whether this profile will be used to establish WAN connections (keyword **out**), to establish inbound connections (keyword **in**), or to establish both (keyword **both**).

cp { *name* | *index* } **telco dn** [**1** | **2**] *string*

This command allows you to set the profile's directory number, or number-to-dial (DN). The number can be up to 32 characters in length and may contain non-dialable characters, which are ignored when placing a call.

Note: For the **cp** and **no cp** versions of this command, if no DN index is specified, **1** is assumed. For the **show cp** version of this command, if no DN index is specified, both DNs will be displayed, each on its own line.

cp { *name* | *index* } **telco prefix** *string*

This command allows you to set the profile's dial prefix. The prefix can be up to three characters long and may contain non-dialable characters, which are ignored when placing a call. The prefix field is prepended to the directory number when placing a call.

Note: This parameter is used ONLY by routers with analog modem interfaces installed.

cp { *name* | *index* } **telco callback** { **yes** | **no** }

This command allows you to configure a profile so that when it is used to accept an incoming call, the router will hang up that call and use its (prefix and) directory number to call back the device that originated the initial call. This is useful when you want a particular party to be billed for WAN connections.

cp { *name* | *index* } **ip nat enable** { **yes** | **no** }

This command allows you to enable or disable Network Address Translation for the profile. Enabling NAT is not sufficient – you must also attach a rule list and optionally a server list using the commands below.

cp { *name* | *index* } **ip nat rule-list** *list-tag*

This command allows you to attach a previously configured IP NAT rule list to a particular profile. *list-tag* should be the name of the desired rule list to use for this profile.

no cp { *name* | *index* } **ip nat rule-list**

This command allows you to detach an IP NAT rule list from a particular profile.

cp { *name* | *index* } **ip nat server-list** *list-tag*

This command allows you to attach a previously configured IP NAT server list to a particular profile. *list-tag* should be the name of the desired server list to use for this profile.

no cp { *name* | *index* } **ip nat server-list**

This command allows you to detach an IP NAT Server List from a particular profile.

show cp { *name* | *index* } **id**

This command displays a connection profile's name and index number. *name* can be any unique descriptive alphanumeric string. *index* can be any value from 1 to 16.

---

cp { *name* | *index* } **connection demand** { **yes** | **no** }

This command allows you to specify whether or not a connection profile will connect “on demand”.

---

cp { *name* | *index* } **connection timeout** *seconds*

This command allows you to specify the idle timeout value in seconds for a connection profile.

PPTP commands

---

cp { *name* | *index* } **pptp ip partner** *ip-addr*

This command allows you to specify a PPTP partner IP address for a particular connection profile specified by *name* or *index*.

---

cp { *name* | *index* } **pptp ip via** *ip-addr*

This command allows you to specify a gateway by which the PPTP partner IP address can be reached when the partner address is in the same subnet as the remote IP address.

If you do not specify the PPTP partner IP address, the router will use the default gateway to reach the partner. If the partner should be reached via an alternate port (i.e. the LAN instead of the WAN), the Tunnel Via Gateway field allows this path to be resolved.

---

cp { *name* | *index* } **pptp authentication type** { **pap** | **chap** | **mschap** }

This command allows you to specify a PPTP authentication type, PAP, CHAP, or MS-CHAP, for a particular connection profile specified by *name* or *index*.

---

cp { *name* | *index* } **pptp compression** { **none** | **standardlzs** }
no cp { *name* | *index* } **pptp compression**

These commands allow you to specify or delete a PPTP compression algorithm, either none or Standard LZS, for a particular connection profile specified by *name* or *index*.

---

cp { *name* | *index* } **pptp encryption** { **none** | **mppe** }
no cp { *name* | *index* } **pptp encryption**

These commands allow you to specify or delete a PPTP encryption algorithm, either none or MPPE, for the specified connection profile.

---

cp { *name* | *index* } **pptp authentication** { **send** | **receive** } **name** *string*
no cp { *name* | *index* } **pptp authentication** { **send** | **receive** } **name**

These commands allow you to set or delete the user name as an alphanumeric string that the specified connection profile will use for PPTP authentication.

---

show cp { *name* | *index* } **pptp authentication** { **send** | **receive** } **name**

This command allows you to show the user name as an alphanumeric string that the specified connection profile uses for PPTP authentication.

---

cp { *name* | *index* } **pptp authentication** { **send** | **receive** } **password** *string*
no cp { *name* | *index* } **pptp authentication** { **send** | **receive** } **password**

These commands allow you to set or delete the password as an alphanumeric string that the specified connection profile will use for PPTP authentication.

Manual connect/disconnect commands

connect cp { *name* | *index* }

Invoking this command with a valid, applicable connection profile will cause the router to attempt to make the appropriate connection, using the profile's settings. A valid, applicable connection profile must be either a profile that matches the primary WAN interface's data link encapsulation, or a tunnel profile.

If the specified profile is valid in this context, the console remains in a modal state until one of the following occurs:

- you type Control-C
- the connection is established, in which case the word "connect" is displayed
- the connection fails, in which case the word "down" is displayed, followed by an appropriate error message

disconnect cp { *name* | *index* }

This command allows you to disconnect the connection, if any, associated with the specified profile. If no connection is in place an error message is displayed. This command returns immediately; the connection disconnect process may still be in progress since it is asynchronous.

Backup configuration commands

cp { *name* / *index* } **interface-group** { **primary** | **backup** | **auxiliary** }
show cp { *name* / *index* } **interface-group**

These commands allow you to set or show the interface group to which the dial backup feature is applied.

Note: **auxiliary** is only allowed if the router is an Ethernet-to-Ethernet router that has the dial-in kit installed.

RIP-2 MD5 configuration commands

cp *id* **ip rip auth key** *id*
no cp *id* **ip rip auth key** *id*
show config cp *id* **ip rip auth key**

These commands allow you to create, delete, or show the RIP-2 Authentication key(s) on the specified Connection Profile.

cp *id* **ip rip auth key** *id* **start date** *date*
show cp *id* **ip rip auth key** *id* **start date**

These commands allow you to set or show a start date for the RIP-2 Authentication key(s) on the specified Connection Profile.

cp *id* **ip rip auth key** *id* **start time** *time*
show cp *id* **ip rip auth key** *id* **start time**

These commands allow you to set or show a start time for the RIP-2 Authentication key(s) on the specified Connection Profile.

---

cp *id* **ip rip auth key** *id* **end date** *date*
show cp *id* **ip rip auth key** *id* **end date**

These commands allow you to set or show an end date for the RIP-2 Authentication key(s) on the specified Connection Profile. The acceptable year range is from 1904 – 2039.

---

cp *id* **ip rip auth key** *id* **end time** *time*
show cp *id* **ip rip auth key** *id* **end time**

These commands allow you to set or show an end time for the RIP-2 Authentication key(s) on the specified Connection Profile.

---

cp *id* **rip auth key** *id* **end time mode** { **infinite** | **date** }
show cp *id* **ip rip auth key** *id* **end time mode**

These commands allow you to set or show the end time mode for the RIP-2 Authentication key(s) on the specified Connection Profile. **date** specifies that an expiration date and time will be used; **infinite** specifies that the key will never expire.

---

cp *id* **rip auth key** *id* **key** <*string*>

These commands allow you to assign a RIP-2 Authentication key on the specified Connection Profile. Keys must be manually entered and must consist of 1 – 16 ASCII characters each.

IP NAT Passthrough Commands

Note: The commands in this section are supported beginning with Firmware Version 8.2.

---

cp { *name* | *index* } **ip nat passthrough enable** { **yes** | **no** }
no cp { *name* | *index* } **ip nat passthrough enable**
show cp { *name* | *index* } **ip nat passthrough enable**

These commands allow you to enable, disable, or show the NAT passthrough behavior for the specified Connection Profile. The IP passthrough feature allows for a single LAN PC to have the router's public address assigned to it, in addition to providing PAT (NAPT) via the same public IP address for all other hosts on the private LAN subnet.

---

cp { *name* | *index* } **ip nat passthrough dhcp enable** { **yes** | **no** }
no cp { *name* | *index* } **ip nat passthrough dhcp enable**
show cp { *name* | *index* } **ip nat passthrough dhcp enable**

These commands allow you to enable, disable, or show the NAT passthrough DHCP behavior for the specified Connection Profile. This governs DHCP addressing for the passthrough host.

---

cp { *name* | *index* } **ip nat passthrough dhcp mac-address** { *mac-address* }
show cp { *name* | *index* } **ip nat passthrough dhcp mac-address**

These commands allow you to set or show the NAT passthrough DHCP MAC address for the specified Connection Profile. This specifies the MAC address of the passthrough host.

Beginning with Firmware Version 8.3.3, IP Passthrough allows a *first come first serve* mode, which defaults to an all-zeroes MAC address.

If you leave the default all-zeroes MAC address, the Router will select the next DHCP client that initiates a DHCP lease request or renewal to be the IP passthrough host. When the WAN comes up, or if it is already up, the Router will serve this client the IP passthrough/WAN address. When this client's lease ends, the IP passthrough address becomes available for the next client to initiate a DHCP transaction. The next client will get the IP passthrough address. Note that there is no way to control which PC has the IP passthrough address without releasing all other DHCP leases on the LAN.

Note: If you specify a non-zeroes MAC address, the DHCP Client Identifier must be in the format specified above. Macintosh computers allow the DHCP Client Identifier to be entered as a name or text, however Motorola Netopia® routers accept only strict (binary/hex) MAC address format. Macintosh computers display their strict MAC addresses in the TCP/IP Control Panel (Classic MacOS) or the Network Preference Pane of System Preferences (Mac OS X).

Once configured, the passthrough host's DHCP leases will be shortened to two minutes. This allows for timely updates of the host's IP address, which will be a private IP address *before* the WAN connection is established. *After* the WAN connection is established and has an address, the passthrough host can renew its DHCP address binding to acquire the WAN IP address.

A restriction

Since both the router and the passthrough host will use same IP address, new sessions that conflict with existing sessions will be rejected by the router. For example, suppose you are a teleworker using an IPSec tunnel from the router *and* from the passthrough host. Both tunnels go to the same remote endpoint, such as the VPN access concentrator at your employer's office. In this case, the first one to start the IPSec traffic will be allowed; the second one – since, from the WAN it's indistinguishable – will fail.

Stateful Inspection Commands

See also:

- "Stateful Inspection Commands" on page 2-85 for Global Stateful Inspection commands.
- "Stateful Inspection Configuration Commands" on page 2-32 for Ethernet interface commands.

Note: The commands in this section are supported beginning with Firmware Version 8.2.

cp { *name* | *index* } **ip state-insp enable** { **yes** | **no** | **on** | **off** }
no cp { *name* | *index* } **ip state-insp enable**
show cp { *name* | *index* } **ip state-insp enable**

These commands allow you to set, disable, or show the status of stateful inspection for the specified Connection Profile. This option is disabled by default. Stateful inspection prevents unsolicited inbound access when NAT is disabled.

cp { *name* | *index* } **ip state-insp xposed-list** *xposed-list_name*
no cp { *name* | *index* } **ip state-insp xposed-list**
show cp { *name* | *index* } **ip state-insp xposed-list**

These commands allow you to set, disable, or show the status of a stateful inspection exposed address list for the specified Connection Profile. Exposed address lists are similar to NAT server lists. Exposed addresses in the list will not be subject to stateful inspection and hence unsolicited inbound traffic will be allowed to these addresses.

These are active only if NAT is disabled on the profile.

cp { *name* | *index* } **ip state-insp tcp-seq-diff** *diff*
show cp { *name* | *index* } **ip state-insp tcp-seq-diff**

These commands allow you to set or show TCP sequence difference acceptable for the specified Connection Profile.

cp { *name* | *index* } **ip state-insp router-access** { **yes** | **no** | **on** | **off** }
no cp { *name* | *index* } **ip state-insp router-access**
show cp { *name* | *index* } **ip state-insp router-access**

These commands allow you to set, disable, or show the status of default mapping to router for the specified Connection Profile.

cp { *name* | *index* } **ip state-insp deny-frag** { **yes** | **no** | **on** | **off** }
no cp { *name* | *index* } **ip state-insp deny-frag**
show cp { *name* | *index* } **ip state-insp deny-frag**

These commands allow you to set, disable, or show whether fragmented packets are received for the specified Connection Profile.

L2TP Connection Profile Configuration Commands

Note: The commands in this section are supported beginning with Firmware Version 8.2.

cp { *name* | *index* } **l2tp ip partner** *ip-addr*
show cp { *name* | *index* } **l2tp ip partner**

These commands allow you to specify or show the partner Ipv4 address associated with the local L2TP tunnel.

cp { *name* | *index* } **l2tp ip via** *ip-addr*
show cp { *name* | *index* } **l2tp ip via**

These commands allow you to specify or show a gateway address at which the partner address may be reached when the partner IP and the remote IP addresses are on the same subnet.

cp { *name* | *index* } **l2tp authentication enable** { **yes** | **no** }
show cp { *name* | *index* } **l2tp authentication enable**

These commands allow you to enable, disable or show the status of L2TP CHAP-like tunnel authentication for the specified profile.

---

cp { *name* | *index* } **l2tp authentication passphrase** *string*

This command sets the local (i.e. LAC/LNS) passphrase. The passphrase must be *at least* eight characters in length. This value is used to establish the shared secret key that must be present when a LAC/LNS pair wants to authenticate an L2TP tunnel.

---

cp { *name* | *index* } **l2tp authentication ppp type** { **pap** | **chap** }
show cp { *name* | *index* } **l2tp authentication ppp**

These commands allow you to specify an L2TP authentication type, PAP or CHAP, for a particular connection profile specified by *name* or *index*.

---

cp { *name* | *index* } **l2tp authentication ppp** { **send** | **receive** } **name** *string*
show cp { *name* | *index* } **l2tp authentication ppp** { **send** | **receive** } **name**

These commands allow you to set or show the user name as an alphanumeric string that the specified connection profile will use for L2TP authentication.

---

cp { *name* | *index* } **l2tp authentication ppp** { **send** | **receive** } **password** *string*
show cp { *name* | *index* } **l2tp authentication ppp** { **send** | **receive** } **password**

These commands allow you to set or show the password as an alphanumeric string that the specified connection profile will use for L2TP authentication.

---

cp { *name* | *index* } **l2tp compression** { **none** | **standardlzs** }
no cp { *name* | *index* } **l2tp compression**
show cp { *name* | *index* } **l2tp compression**

These commands allow you to specify, disable, or show the PPP compression algorithm, if any. The current options are **none** (for no compression) and **standardlzs** for compression.

GRE Connection Profile Configuration Commands

Note: The commands in this section are supported beginning with Firmware Version 8.4.

---

cp { *name* | *index* } **gre ip partner** *ip-addr*
show cp { *name* | *index* } **gre ip partner**

These commands allow you to specify or show the partner IPv4 address associated with the local GRE tunnel connection profile.

---

cp { *name* | *index* } **gre ip via** *ip-addr*
show cp { *name* | *index* } **gre ip via**

These commands allow you to specify or show the gateway (next hop forwarding) address used when routing GRE (tunneled) packets to the partner. when the partner IP and the remote IP addresses are on the same subnet.

cp { *name* | *index* } **gre checksum** [**yes** | **no**]
show cp { *name* | *index* } **gre checksum** [**yes** | **no**]

These commands allow you to enable, disable, or show whether a GRE tunnel will transmit a checksum field on outgoing GRE packets, when enabled. The Router will implicitly check incoming GRE packets with a checksum value.

cp { *name* | *index* } **gre sequence-datagrams** [**yes** | **no**]
show cp { *name* | *index* } **gre sequence-datagrams** [**yes** | **no**]

These commands allow you to enable, disable, or show whether a GRE tunnel will transmit sequence numbers on outgoing GRE packets, when enabled. Also specifies that the GRE tunnel will check an incoming GRE packet's sequence number field for out-of-sequence ordering and buffering. When disabled, the CPE will not send sequence numbers; and it will consider *all* inbound packets to be in-sequence (i.e. not check the "S" bit flag, if set).

cp { *name* | *index* } **gre key** [0..232-1]
show cp { *name* | *index* } **gre key** [0..232-1]

These commands allow you to specify or show a 32-bit integer key value assigned to the GRE tunnel; if zero, the tunnel is considered not to have a key identifier. It will not set the "K" bit in the GRE header, and ignore any key value received from the peer. If non-zero, the received key field value (if present in the GRE header) will be checked against the local tunnel value, and discarded if the comparison fails. If non-zero, all outbound GRE headers will include this value in the Key Field.

CompuServe Login

CompuServe Login Connection Profile Commands
cp { *name* | *index* } **telco compuserve login** { **yes** | **no** } **show cp** { *name* | *index* } **telco compuserve login** **no cp** { *name* | *index* } **telco compuserve login** **cp** { *name* | *index* } **telco compuserve hostname** *string* **show cp** { *name* | *index* } **telco compuserve hostname** **no cp** { *name* | *index* } **telco compuserve hostname** **cp** { *name* | *index* } **telco compuserve username** *string* **show cp** { *name* | *index* } **telco compuserve username** **no cp** { *name* | *index* } **telco compuserve username** **cp** { *name* | *index* } **telco compuserve password** *string* **no cp** { *name* | *index* } **telco compuserve password**

---

cp { *name* | *index* } **telco compuserve login** { **yes** | **no** }
show cp { *name* | *index* } **telco compuserve login**
no cp { *name* | *index* } **telco compuserve login**

These commands set, display, or disable the specified connection profile's CompuServe login enable setting.

---

cp { *name* | *index* } **telco compuserve hostname** *string*
show cp { *name* | *index* } **telco compuserve hostname**
no cp { *name* | *index* } **telco compuserve hostname**

cp { *name* | *index* } **telco compuserve username** *string*
show cp { *name* | *index* } **telco compuserve username**
no cp { *name* | *index* } **telco compuserve username**

cp { *name* | *index* } **telco compuserve password** *string*
no cp { *name* | *index* } **telco compuserve password**

These commands set, display, or disable the specified connection profile's CompuServe login host-name, user-name, or password string. For security reasons, there is no show variant of the **cp** { *name* | *index* } **telco compuserve password** command.

IPSec/IKE

Connection Profile IPSec Configuration Commands
cp { *name* | *index* } **ipsec suite encryption** { **des** | **3des** | **null** } **authentication** { **esp** | **ah** } { **md5** | **sha1** } [**compression** { **none** | **lzs** }]
cp { *name* | *index* } **ipsec ip** [**remote** {[**members** {*xxx.xxx.xxx.xxx/nn* | *xxx.xxx.xxx.xxx xxx.xxx.xxx.xxx*}] [**tep** *x.x.x.x*]}] [**local tep** *x.x.x.x*] [**via** *x.x.x.x*]
cp { *name* | *index* } **ipsec ip [modify** *net-index* **]** **remote members** *remote IPv4-addr1* **[** */nn* **|** *remote IPv4-addr2* **]** **local members** *local IPv4-addr1* **[** */ nn* **|** *local IPv4-addr2* **]**
cp { *name* | *index* } **ipsec spi** *rx-esp-spi* [*tx-esp-spi* [*rx-ah-spi* [*tx-ah-spi*]]]
cp { *name* | *index* } **ipsec authentication key** *string*
cp { *name* | *index* } **ipsec encryption key** *1234567890123456* [*1234567890123456 1234567890123456*]

---

cp { *name* \| *index* } **ipsec suite encryption** { **des** \| **3des** \| **null** }
authentication { **esp** \| **ah** } { **md5** \| **sha1** } [**compression** { **none** \| **lzs** }]

This command allows you to specify the IPsec suite encryption type and authentication method for an IPsec tunnel.

---

cp { *name* \| *index* } **ipsec ip**
[**remote** {[**members** {*xxx.xxx.xxx.xxx/nn* \| *xxx.xxx.xxx.xxx xxx.xxx.xxx.xxx*}] [**tep** *x.x.x.x*]}]
[**local tep** *x.x.x.x*] [**via** *x.x.x.x*]

This command sets all the pertinent IP values for the IPSec tunnel. There are three sub-sections of this command, the **remote**, **local**, and **via**. The **remote** section, if it exists, may contain a **members** or a **tep** (“tunnel endpoint”) parameter, or both. The **local** section, if it exists, may contain only a **tep** parameter. The optional **via** section sets the next hop gateway.

---

cp { *name* \| *index* } **ipsec ip [modify** *net-index* **]**
remote members *remote IPv4-addr1* **[** */nn* **\|** *remote IPv4-addr2* **]**
local members *local IPv4-addr1* **[** */ nn* **\|** *local IPv4-addr2* **]**

Note: This command is supported beginning with firmware release 8.2.

This command sets or modifies all the pertinent IP values for the IPSec tunnel:

- *remote IPv4-addr1* – lowest IPv4 address in the remote network
- *remote IPv4-addr2* – highest IPv4 address in the remote ranged network
- *local IPv4-addr1* – lowest IPv4 address in the local network
- *local IPv4-addr2* – highest IPv4 address in the local ranged network
- *nn* – number of bits in the subnet mask (*nn* = 0...31). Current network is defined as a subnet.
- *net-index* – a configured network's internal ordinal index, which is known by performing the **show** command, shown below.

If a **remote members** or **local members** option (**[** *foo* **]**) is not present, the network is defined as a host net (single address).

The **tep** clause sets the remote tunnel endpoint IP. In order for the profile to function properly, it must be specified once. It becomes an optional clause thereafter.

If the **modify** option is absent, it means a *new* network configuration is added to the config area. Its presence indicates a *change* to an existing network.

Examples:

Change a tunnel's remote tunnel endpoint:

```
cp { name | index } ipsec ip remote members x.x.x.x
```

Change a tunnel's local tunnel endpoint:

```
cp { name | index } ipsec ip local members x.x.x.x
```

Change a tunnel's via gateway:

```
cp { name | index } ipsec ip via x.x.x.x
```

Display a Connection Profile's List of Network Configurations:

```
show config cp { name | index }
```

Delete a Network within a Connection Profile, or All Networks:

```
no cp { name | index } ipsec ip network { net-index | all }
```

---

cp { *name* / *index* } **ipsec spi** *rx-esp-spi* [*tx-esp-spi* [*rx-ah-spi* [*tx-ah-spi*]]]

This command allows you to specify the security parameters indexes for an IPsec tunnel.

---

cp { *name* / *index* } **ipsec authentication key** *string*

This command allows you to specify the authentication secret for an IPsec tunnel. You must specify an authentication secret if the authentication type is anything other than None.

Note: The key is a hexadecimal entry of 16 bytes (32 characters of input) for **md5** and 20 bytes (40 characters of input) for **sha1**. It is not possible to retrieve the encryption keys or authentication key once they have been set.

cp { *name* | *index* } **ipsec encryption key** *1234567890123456* [*1234567890123456 1234567890123456*]

This command allows you to specify the authentication key for an IPsec tunnel. You must specify an authentication key if the authentication type is anything other than None. The key must be an ASCII string of up to 48 characters for both **md5** and **sha1**.

Note: For DES the key is one group of 16 hexadecimal characters; for 3DES the key is three groups of 16 hexadecimal characters each.

IKE/IPSec Connection Profile Commands

cp { *name* | *index* } **ipsec dead-peer-detection** { **yes** | **no** }
show cp { *name* | *index* } **ipsec dead-peer-detection**
no cp { *name* | *index* } **ipsec dead-peer-detection**

cp { *name* | *index* } **ipsec dead-peer-detection ping-address** *remote_net_IPv4_address*
show cp { *name* | *index* } **ipsec dead-peer-detection ping-address**

cp { *name* | *index* } **ipsec dead-peer-detection ping-retry** *1..65535*
show cp { *name* | *index* } **ipsec dead-peer-detection ping-retry**

cp { *name* | *index* } **ipsec dead-peer-detection ping-reply-timeout** *1..65535*
show cp { *name* | *index* } **ipsec dead-peer-detection ping-reply-timeout**

cp { *name* | *index* } **ipsec idle-timeout** { *non-negative-integer* | **none** }
show cp { *name* | *index* } **ipsec idle-timeout**
no cp { *name* | *index* } **ipsec idle-timeout**

cp { *name* | *index* } **ipsec key-manager** { **manual** | **ike** }
show cp { *name* | *index* } **ipsec key-manager**

cp { *name* | *index* } **ipsec ike phase1** { *name* | *index* | **none** }
show cp { *name* | *index* } **ipsec ike phase1**
no cp { *name* | *index* } **ipsec ike phase1**

cp { *name* | *index* } **ipsec pfs** { **yes** | **no** }
show cp { *name* | *index* } **ipsec pfs**
no cp { *name* | *index* } **ipsec pfs**

cp { *name* | *index* } **ipsec suite encapsulation** { **esp** | **ah** | **esp+ah** }
[**encryption** { **des** | **3des** | **null** }]
[**authentication esp** { **md5** | **hmac-md5-96** | **sha1** | **hmac-sha1-96** }]
[**authentication ah** { **md5** | **hmac-md5-96** | **sha1** | **hmac-sha1-96** }]
[**compression lzs**]
show cp { *name* | *index* } **ipsec suite**

IKE/IPSec Connection Profile Commands
cp { *name* | *index* } **ipsec ip** [**remote** [**members** {a.b.c.d | a.b.c.d/n | a.b.c.d e.f.g.h | a.b.c.d-e.f.g.h}] [**tep** a.b.c.d]] [**local** [**members** {a.b.c.d | a.b.c.d/n | a.b.c.d e.f.g.h | a.b.c.d-e.f.g.h}] [**tep** a.b.c.d]] [**via** a.b.c.d] **show cp** { *name* | *index* } **ipsec ip**
cp { *name* | *index* } **ipsec sa lifetime** { **seconds** | **kbytes** } { *non-negative-integer* | **none** } **show cp** { *name* | *index* } **ipsec sa lifetime** [{ **seconds** | **kbytes** }] **no cp** { *name* | *index* } **ipsec sa lifetime** [{ **seconds** | **kbytes** }]

cp { *name* \| *index* } **ipsec dead-peer-detection** { **yes** \| **no** }
show cp { *name* \| *index* } **ipsec dead-peer-detection**
no cp { *name* \| *index* } **ipsec dead-peer-detection**

These commands set, display, or disable the status of dead peer detection for the specified IPsec Phase 2 profile. Dead peer detection counts the outbound packets on a tunnel. If 256 packets go out without a single packet coming in, the tunnel SAs are expired and a rekey is started. Rekeying is first attempted on the previous Phase 1 SA. If the Phase 1 request times out, then the Phase 1 SA is expired and Phase 1 rekeying is begun over again.

cp { *name* \| *index* } **ipsec dead-peer-detection ping-address** *remote_net_IPv4_address*
show cp { *name* \| *index* } **ipsec dead-peer-detection ping-address**

Note: These commands are supported beginning with firmware version 8.2

These commands allow you to specify or show what IP destination host address is used to verify whether or not peer is dead. The IP address *must* belong to a tunnel's remote network (which can be configured as a subnet, an address range, or an individual host in the IP options menu). The subnet remote network case also disallows the host part of the address to be all ones or all zeroes. For example, it is not permitted to set the address to 163.176.0.0 or 163.176.255.255 in a class B network.

cp { *name* \| *index* } **ipsec dead-peer-detection ping-retry** *1..65535*
show cp { *name* \| *index* } **ipsec dead-peer-detection ping-retry**

Note: These commands are supported beginning with firmware version 8.2

These commands allow you to specify or show the retry interval between successive pings (in seconds). Default is 5 seconds.

---

cp { *name* | *index* } **ipsec dead-peer-detection ping-reply-timeout** *1..65535*
show cp { *name* | *index* } **ipsec dead-peer-detection ping-reply-timeout**

Note: These commands are supported beginning with firmware version 8.2

These commands allow you to specify or show the maximum period of time (in seconds) an IPsec tunnel endpoint will wait for the peer's response to its earliest ping request. If the peer does not respond within this period, it is deemed to be a dead peer tunnel. Default is 90 seconds.

---

cp { *name* | *index* } **ipsec idle-timeout** { *non-negative-integer* | **none** }
show cp { *name* | *index* } **ipsec idle-timeout**
no cp { *name* | *index* } **ipsec idle-timeout**

These commands set or display the idle timeout associated with the specified IPSec connection profile. If the IPSec **key-manager** associated with the connection profile is **manual**, then the idle-timeout value is meaningful only if the **remote sg** is 0.0.0.0 or the empty string. In that case, the idle-timeout value specifies the period in seconds during which the SPI (or SPIs) are bound to a particular remote peer in the absence of outbound traffic through the IPSec tunnel. The value zero (or the keyword **none**) causes the SPI (or SPIs) to be permanently bound to the first remote peer that sends traffic through the tunnel using the SPI (or SPIs). If the IPSec **key-manager** associated with the connection profile is **ike**, then the idle-timeout value specifies the period prior to SA expiration during which there must be at least one outbound packet through the IPSec tunnel for a re-key to be performed one second prior to SA expiration. The value zero (or the keyword **none**) indicates that a re-key should always be performed one second prior to SA expiration even if there has been no outbound traffic through the tunnel.

---

cp { *name* | *index* } **ipsec key-manager** { **manual** | **ike** }
show cp { *name* | *index* } **ipsec key-manager**

These commands set or display the IPSec key manager associated with the specified connection profile.

---

cp { *name* | *index* } **ipsec ike phase1** { *name* | *index* | **none** }
show cp { *name* | *index* } **ipsec ike phase1**
no cp { *name* | *index* } **ipsec ike phase1**

These commands set, display, or disable the IKE Phase1 profile associated with the specified connection profile. The IKE Phase1 profile may be specified either by index or by name.

---

cp { *name* | *index* } **ipsec pfs** { **yes** | **no** }
show cp { *name* | *index* } **ipsec pfs**
no cp { *name* | *index* } **ipsec pfs**

These commands set, display, or change the Phase 2 perfect forward secrecy setting for the specified IPsec Phase 2 profile.

cp { *name* | *index* } **ipsec suite encapsulation** { **esp** | **ah** | **esp+ah** }
[**encryption** { **des** | **3des** | **null** }]
[**authentication esp** { **md5** | **hmac-md5-96** | **sha1** | **hmac-sha1-96** }]
[**authentication ah** { **md5** | **hmac-md5-96** | **sha1** | **hmac-sha1-96** }]
[**compression lzs**]
show cp { *name* | *index* } **ipsec suite**

Note: This is an extended version of an existing CLI command. The existing command is modified to add an encapsulation clause and to allow for one or two authentication clauses. See "IPSec/IKE" on page 3-26 for more information.

These commands set or display the IPSec encapsulation, encryption, authentication, and compression parameters for the specified connection profile.

Note: The authentication clause may appear either one or two times; if it appears twice, one occurrence must specify ah and the other must specify esp.

The keywords **md5** and **hmac-md5-96** are synonyms, although the latter keyword is preferred, the former being retained only for backwards compatibility. The keywords **sha1** and **hmac-sha1-96** are synonyms, although the latter keyword is preferred, the former being retained only for backwards compatibility.

cp { *name* | *index* } **ipsec ip**
[**remote**
[**members** {a.b.c.d | a.b.c.d/n | a.b.c.d e.f.g.h | a.b.c.d-e.f.g.h}]
[**tep** a.b.c.d]]
[**local**
[**members** {a.b.c.d | a.b.c.d/n | a.b.c.d e.f.g.h | a.b.c.d-e.f.g.h}]
[**tep** a.b.c.d]]
[**via** a.b.c.d]
show cp { *name* | *index* } **ipsec ip**

Note: This is an extended version of an existing CLI command. The existing command is modified to allow a members specification to appear in the local clause and to allow for a host address or an IP address range (rather than a network address and subnet mask) in the remote and local members clauses. See "IPSec/IKE" on page 3-26 for more information.

This command sets the pertinent IP values for the IPSec tunnel, and may contain zero or one instances of each of three possible clauses: **remote**, **local**, and **via**. The **remote** clause, if specified, may include a members specification or a tunnel endpoint ("tep") specification, or both. The **local** clause, if specified, may contain a members specification or a tunnel endpoint specification, or both. The optional **via** clause sets the next hop gateway. The keyword **sg** (short for "security-gateway") is an acceptable synonym for the keyword **tep**.

cp { *name* | *index* } **ipsec sa lifetime** { **seconds** | **kbytes** } { *non-negative-integer* | **none** }
show cp { *name* | *index* } **ipsec sa lifetime** [{ **seconds** | **kbytes** }]
no cp { *name* | *index* } **ipsec sa lifetime** [{ **seconds** | **kbytes** }]

These commands set, display, or disable one or both of the two IKE Phase 2 SA lifetimes (in seconds and/or kbytes protected) for the specified IPSec protocol for the specified connection profile. Specifying neither the keyword **seconds** nor the keyword **kbytes** with the show variant of this command displays both lifetime values. The keyword **none** is equivalent to the value zero, and indicates that there is no lifetime of the specified type.

Note: It is a run-time checked error if both of the IKE Phase 2 SA lifetime values for a particular protocol are set to zero or **none**.

ICMP Dead Peer Detection Commands

Beginning with the version 8.2 firmware release, the Command Line Interface supports the following new and modified Connection Profile configuration commands:

IKE/IPSec Dead Peer Detection Connection Profile Commands
cp { *name* | *index* } **ipsec dead-peer-detection enable** { **yes** | **no** } **no cp** { *name* | *index* } **ipsec dead-peer-detection** **show cp** { *name* | *index* } **ipsec dead-peer-detection enable** **cp** { *name* | *index* } **ipsec dead-peer-detection ping-address** *remote net IPv4 address* **show cp** { *name* | *index* } **ipsec dead-peer-detection ping-address** **cp** { *name* | *index* } **ipsec dead-peer-detection ping-retry** *1...65535* **show cp** { *name* | *index* } **ipsec dead-peer-detection ping-retry** **cp** { *name* | *index* } **ipsec dead-peer-detection ping-reply-timeout** *1...65535* **show cp** { *name* | *index* } **ipsec dead-peer-detection ping-reply-timeout**

cp { *name* \| *index* } **ipsec dead-peer-detection enable** { **yes** \| **no** }
no cp { *name* \| *index* } **ipsec dead-peer-detection**
show cp { *name* \| *index* } **ipsec dead-peer-detection enable**

These commands allow you to enable, disable, or show the status of the ICMP Dead Peer Detection feature. The **no cp...** command is equivalent to specifying the **no** option.

cp { *name* \| *index* } **ipsec dead-peer-detection ping-address** *remote net IPv4 address*
show cp { *name* \| *index* } **ipsec dead-peer-detection ping-address**

These commands allow you to specify or show the IP destination host address that will be used to verify if the peer is dead or not. The IP address ***must*** belong to a tunnel's remote network. A tunnel's remote network can be configured as a subnet, an address range, or an individual host. The subnet remote network case also disallows the host part of the address to be all ones or all zeroes. For example, the addresses 163.176.0.0 or 163.176.255.255 are not permitted in a class B network.

cp { *name* \| *index* } **ipsec dead-peer-detection ping-retry** *1...65535*
show cp { *name* \| *index* } **ipsec dead-peer-detection ping-retry**

These commands allow you to specify or show the retry interval between successive pings (in seconds). Default is 5 seconds.

cp { *name* | *index* } **ipsec dead-peer-detection ping-reply-timeout** *1...65535*
show cp { *name* | *index* } **ipsec dead-peer-detection ping-reply-timeout**

These commands allow you to specify or show the maximum period of time (in seconds) an IPsec tunnel endpoint should wait for the peer's response to its earliest ping request. If the peer does not respond within this period, it is deemed to be a dead peer tunnel. Default is 90 seconds.

IPSec MTU Command

Beginning with Version 8.4 firmware, the Command Line Interface supports the following new Connection Profile configuration command:

IPSec MTU Connection Profile Command
cp [*name* | *index*] **ipsec mtu** *value* **show cp** [*name* | *index*] **ipsec mtu**

cp [*name* | *index*] **ipsec mtu** *value*
show cp [*name* | *index*] **ipsec mtu**

These commands allow you to specify or show a manual maximum transmission unit (MTU) – also called Maximum Packet Size – parameter for the specified Connection Profile. The maximum value (also the default) is 1500, and the minimum is 100.

This is the starting value that is used for the MTU when the IPSec tunnel is installed. It specifies the maximum IP packet length for the encapsulated AH or ESP packets sent by the router. The MTU used on the IPSec connection will be automatically adjusted based on the MTU value in any received ICMP *can't fragment* error messages that correspond to IPSec traffic initiated from the router. Normally the MTU only requires manual configuration if the ICMP error messages are blocked or otherwise not received by the router.

Chapter 4

Motorola Netopia® Router Text Configuration Upload

This chapter describes the supported TFTP text configuration upload process.

TFTP Text Configuration Upload Overview

You can configure many of the basic features of the router by uploading a text-based configuration file to the router. This file can be either a Macintosh- or PC-formatted text file. There must be no formatting information in the file – it must contain only raw text. Generally this means that you must save the file in Text Only (.txt) format when using word processing applications that support text formatting.

The file must be located on a TFTP Server. The Motorola Netopia® router needs the IP Address or DNS Name of the TFTP Server in order to start the file upload. There are at least three ways to accomplish this:

- SNMP
- VT100 Menu Console (Serial or Telnet)
- VT100 Command Line Console (Serial or Telnet)

The supported character set for TFTP text configuration files is the set of US-ASCII printable characters (ASCII values from 32 to 126 inclusive), including the space character. This means that characters containing diacritical marks, such as 'À', are not supported. Such characters will be translated to the character '%' when processing a text configuration file.

Note: All commands, including the last, must be followed by an appropriate end-of-line sequence (Carriage Return, Line Feed, or Carriage Return/Line Feed pair).

SNMP

Three SNMP objects are associated with TFTP text configuration upload. They are tftpServerName, tftpConfigFileName, and tftpReadConfig. All three objects are defined in the Netopia MIB. Refer to this document found on the Netopia ftp site for more information.

VT100 Menu Console

The path to the TFTP menu is:

You will need to set the TFTP Server Name (IP address or domain name) and the Config File Name, and then invoke the transfer using GET CONFIG FROM SERVER.

VT100 Command Line Console

The router's console user interface comes up in Menu mode by default. In this mode you use the arrow, Escape, and Return/Enter keys to navigate through a series of screens. To invoke the command line at any time, hit Control-N. The console will erase the window, and you will be presented with a # prompt. The procedure for invoking the file transfer via the command line console is described in the section "Miscellaneous Commands" on page 2-103. To return to Menu mode hit Control-N again.

A command that contains a syntax error will cause the configuration file processing to terminate. Any valid commands that were processed before the error was detected will modify the configuration of the router. The error will be reported in the Device Event History, which will display as much of the text of the offending command as possible.

Example Text Configuration File

The following text file is provided for your use as an example. Make your own appropriate substitutions.

Note: All commands, including the last, must be followed by an appropriate end-of-line sequence (Carriage Return, Line Feed, or Carriage Return/Line Feed pair).

```
;Example config file

;LAN configuration
;set ethernet address
interface ethernet 0 ip address 163.176.227.1/24

;set a secondary ethernet address
interface ethernet 0 ip address 163.176.254.1/24 secondary

;set default gateway to 163.176.224.1
ip gateway 163.176.224.1

;set Rx and Tx RIP
interface ethernet 0 ip rip receive both
interface ethernet 0 ip rip transmit v1

;set dns 1 and dns 2
ip dns 1 163.176.4.10
ip dns 2 163.176.4.31

;set domain name
ip domain-name netopia.com

;configure IP address serving to serve 100 addresses off the 163.176.227.0/24 subnet
ip address-serve 163.176.227.101 163.176.227.201 dhcp
ip address-serve mode server

;other possible commands include:
;interface wan 1 dle ppp
;interface wan 1 dle rfc1483
;user mysecurname mysecurpass

;WAN configuration
;instantiate profile 1 with name "My ISP" and DLE type RFC1483
cp 1
cp 1 tag "My ISP"

;set up profile addressing
cp 1 ip address local 163.176.224.2


;other possible commands include:
;cp 2
;cp 16
;cp 1 ip nat enabled no
```

```
;cp 1 ip nat enabled yes
;cp 1 ip addressing unnumbered
;cp 1 ip addressing numbered
;cp 1 ip address remote 163.176.224.1
;cp 1 ip mask local 255.255.255.0
;cp 1 ip mask remote 255.255.255.0
;cp 1 dle rfc1483
;cp 1 dle ppp
;cp 1 ppp authentication pap
;cp 1 ppp authentication chap
;cp 1 ppp authentication send name "My Name"
;cp 1 ppp authentication send password "My Password"
```

Chapter 5

CLI Error Messages

This chapter describes the CLI error messages and their meaning.

Negative errors

Negative errors are fatal. They will terminate processing of a TFTP configuration file upload if the command that caused the error was executed as part of a TFTP configuration file upload.

Fatal system errors

; error -1: unknown error

This error indicates that an internal error occurred within the command line processor. This error should not occur under normal circumstances.

; error -2: memory allocation failed

This error indicates that the command line processor ran out of memory attempting to complete the requested operation. This error should not occur under normal circumstances.

; error -3: set operation failed

This error indicates that the command line processor failed attempting to complete the requested set operation. This error should not occur under normal circumstances.

Parsing or tokenizing errors

; error -10: input too long

This error indicates that the input exceeds the maximum allowable length for the input field.

; error -11: unexpected end of input

This error indicates that input was not the expected length for the input field.

; error -12: unterminated quoted text

This error indicates that the quoted text input did not have terminating quotes as expected.

Fatal syntax errors

; error -101: no match

This error indicates that you entered an unrecognized command.

; error -120: syntax error

This error indicates that you entered a command with a syntactic error for which the command line processor was unable to provide a more specific error message. For example, you may have misspelled or omitted a keyword or interchanged two keywords.

Example:

```
#frame-relay dlci 22 default
; error -120: syntax error
```

; error -121: illegal operation

This error indicates that you attempted to perform an unsupported operation, or one that does not make sense.

Example:

```
#no version
; error -121: illegal operation
#clear version
; error -121: illegal operation
```

; error -122: illegal value

This error indicates that you entered a properly formatted command, but that one of the values specified in the command is not a valid value for the attribute you are trying to set. For example, this error would be generated if you entered the interface intf-type id dle command and requested a datalink encapsulation that isn't supported by the specified wan interface.

Example:

```
#interface sdsl 1 pvc 0 65536
; error -122: illegal value
```

; error -123: illegal ip address

This error indicates that you supplied an invalid value where an IP address is required. IP addresses should be specified in "dotted-quad" notation: four decimal values, each between 0 and 255 inclusive, separated by dots (e.g., 192.168.1.1).

Example:

```
#interface ethernet 0 ip address xyz
; error -123: illegal ip address
#interface ethernet 0 192.168.256.1/24
; error -123: illegal ip address
```

; error -124: illegal ip mask

This error indicates that you supplied an invalid value where an IP mask is required. An IP mask always may be entered in "dotted-quad" notation – four decimal values, each between 0 and 255 inclusive, separated by dots (e.g., 255.255.255.0). Whenever the IP mask is being entered in conjunction with an IP address, it may also be entered in "prefix" notation – a slash ("/") immediately following the IP address, followed by a value between 0 and 32 inclusive indicating the number of contiguous ones-bits in the mask (e.g., /24). Note that IP mask values entered in dotted-quad notation must consist of a contiguous number of ones-bits beginning with the most significant bit. IP masks with discontiguous ones-bits (such as 255.0.255.0) are invalid.

Example:

```
#int e0 ip address 163.176.1.1/40
; error -124: illegal ip mask
```

; error -125: invalid index

This error indicates that the command referenced a currently non-existent instance of an indexed object. The specified index value might be valid at some point if the appropriate instance of the object were created.

Example:

```
#show frame-relay dlci 22
; error -125: invalid index
#frame-relay dlci 22 tag "My DLCI"
#show frame-relay dlci 22
frame-relay dlci 22 tag "My DLCI" ip-address 0.0.0.0 cir
default bc default be default enable
```

; error -126: number required

This error indicates that a non-numeric value was entered where a numeric one was required.

Example:

```
#ping 192.168.1.1 count abc
; error -126: number required
```

; error -127: index out of bounds

This error indicates that an out of range value was specified in a command that requires an index, such as attempting to access connection profile 17 on a router that supports only sixteen connection profiles. The value supplied is never a valid index value in the context in which it was used.

Example:

```
#sh cp 17
; error -127: index out of bounds
```

; error -128: 'yes' or 'no' required

This error indicates that a value other than 'yes' or 'no' was entered where only 'yes' or 'no' are acceptable.

Example:

```
#cp 1 enable foo
; error -128: 'yes'or 'no' required
```

; error -129: text too long

This error indicates that a string value was entered that was longer than the permissible length for the particular string.

Example:

```
#cp 1 tag "A overly long connection profile name"
; error -129: text too long
```

; error -130: text can't be empty

This error indicates that an empty string was supplied for a string item that must contain at least one character, such as a connection profile name.

Example:

```
#cp 1 tag ""
; error -130: text can't be empty
```

; error -131: invalid text

This error indicates that an invalid string was supplied for a string item that must contain at least one required character.

; error -132: invalid dotted string

This error indicates that an invalid dotted string was supplied for a string item that must contain characters in a standard dotted format, such as an IP address.

; error -133: invalid address/mask

This error indicates either than you omitted a required IP address and/or mask.

Example:

```
#interface ethernet 0 ip address 192.168.1.2
; error -133: invalid address/mask
```

; error -134: invalid keyword

This error indicates either than you entered a keyword that is not a permissible or accepted keyword.

; error -135: missing required text

This error indicates that you omitted a portion of the command.

Example:

```
#ip route
; error -135: missing required text
```

; error -136: illegal or conflicting range

This error indicates that you entered an invalid or an inconsistent range of values. For example, you would receive this error if you entered a range of IP addresses with a starting address that is greater than the ending address.

Example:

```
#ip addr 163.176.12.100 163.176.12.50
; error -136: illegal or conflicting range
```

; error -137: no arp cache entry

This error indicates that you attempted to delete a non-existent arp cache entry.

Example:

```
#arp 192.168.1.1 00:00:C5:70:00:04
#no arp 192.168.1.2 00:00:C5:70:00:04
; error -137: no arp cache entry
```

; error -138: invalid wan port

This error indicates that you entered an invalid interface index. See "Interface Naming Conventions" on page 1-3.

Example:

```
#arp 192.168.1.1 00:00:C5:70:00:04 3
; error -138: invalid wan port
```

; error -139: conflicting duplicate value

This error indicates that you attempted to enter a duplicate value where the same value entry is not permitted.

; error -140: illegal hardware address

This error indicates that you entered an improperly formatted hardware address. The format for an Ethernet MAC address is six hexadecimal values between 00 and FF inclusive separated by colons (e.g., 00:00:C5:70:00:04).

Example:

```
#arp 192.168.1.1 00:00:C5:70:00:
; error -140: illegal hardware address
```

; error -141: no route to the specified IP address

This error indicates that you entered an IP address that is either improperly formatted or cannot be reached.

; error -142: address and mask required

This error indicates that you omitted an IP address and/or mask where both are required.

Example:

```
#no interface ethernet 0 ip address 192.168.1.1
; error -142: address and mask required
```

; error -143: can't add

This error indicates that you attempted to add more than the allowed number of some object.

Example:

```
#interface ethernet 0 address-serve helper 10.0.0.1
#interface ethernet 0 address-serve helper 20.0.0.1
#interface ethernet 0 address-serve helper 30.0.0.1
#interface ethernet 0 address-serve helper 40.0.0.1
#interface ethernet 0 address-serve helper 50.0.0.1
; error -143: can't add
```

; error -144: incomplete command

This error indicates that you omitted a portion of the command.

Example:

```
#interface ethernet 0 ip address
; error -144: incomplete command
```

; error -145: ip address required

This error indicates that you failed to supply an IP address where one is required.

Example:

```
#cp 1 ip address remote
; error -145: ip address required
```

; error -146: ip mask required

This error indicates that you failed to supply an IP mask where one is required.

Example:

```
#cp 1 ip mask remote
; error -146: ip mask required
```

; error -147: too many tokens

This error indicates that you entered more items than are allowed as part of the command. This could result from failing to quote a string value that contains one or more spaces.

Example:

```
#cp 1 telco dn 555 1212
; error -147: unexpected text following command
#cp 1 telco dn "555 1212"
#sh cp 1 telco dn
cp 1 telco dn "555 1212"
#
```

; error -148: text too short

This error indicates that you supplied a string value that is shorter than the minimum permissible length for a string item.

; error -149: no such subnet

This error indicates that you specified (or implicitly referenced) an unknown subnet.

Example:

```
#show interface ethernet 0 ip address
interface ethernet 0 ip address 192.168.1.1/24
#interface ethernet 0 address-serve range 192.168.1.128 192.168.1.159
#interface ethernet 0 address-serve gateway 192.168.2.1
; error -149: no such subnet
```

; error -150 not allowed for this particular item

This error indicates that you attempted an operation that might succeed on a different item in the same array but isn't valid for the specified item.

; error -151 no item matching name exists

This error indicates that you specified an item when no item with that name exists.

; error -152 hex digits only, 0-9 or a-f or A-F

This error indicates that you entered a string for an item that permits only hexadecimal entries, i.e. 0–9 or a–f or A–F.

; error -153 values consisting of all asterisks are ignored

This error is generated if you try to set passwords, keys, or secrets with a value consisting entirely of asterisks. This is to prevent a user from pasting the results of the **show config** command in an attempt to substitute your passwords with asterisks.

; error -160 Key must be exactly 16 hex digits

This error is generated if you try to set keys that require 16 hex digits using more or fewer than 16.

; error -161 DES key must be exactly 16 hex digits

This error is generated if you try to set a DES key, which requires 16 hex digits, using more or fewer than 16.

; error -162 3DES Keys must be 3 keys of exactly 16 hex digits each

This error is generated if you try to set 3DES keys, which requires 16 hex digits each, using more or fewer than 16; or using more or fewer than 3 keys.

; error -163 SHA1 Key must be exactly 40 hex digits

This error is generated if you try to set a SHA1 key, which requires 40 hex digits, using more or fewer than 40.

; error -164 MD5 Key must be exactly 32 hex digits

This error is generated if you try to set an MD5 key, which requires 32 hex digits, using more or fewer than 32.

; error -170 Unupported or invalid Time Zone value

This error is generated if you try to set an invalid Time Zone.

; error -200: execution failed

This error indicates that a requested operation, such as a TFTP configuration file upload, failed.

Example:

```
#receive tftp config 192.168.1.1 myconfig.txt
; error -200: execution failed
```

Voice command errors

```
; error -250: bad extension number

; error -251: extension number does not exist

; error -252: extension number already exists

; error -253: wrong Auto-Attendant time

; error -254: directory number doesn't exist

; error -255: port has not extension number,set phonemap first

; error -256: Directory is full,can't add new one

; error -257: Caller ID list is full

; error -258: Bad Caller ID

; error -259: No such carrier name

; error -260: Carrier Name too long

; error -261: prefix Name too long

; error -262 Pin Name too long

; error -263: Carrier table is full,can't add new one

; error -264: No such dialed digits

; error -265: Route table is full,can't add new one

; error -266: Duplicated dialed string

; error -267: Dialed digits too long
```

; error -268: invalid digit

; error -269: only single digit allowed

Fatal access control errors

; error -400: access denied

This error indicates that you attempted to display an attribute that may not be displayed, or to change an attribute to which you do not have access.

Example:

```
#show cp 1 ppp authentication send password
; error -400: access denied
```

Positive errors

Positive (non-fatal) errors do not terminate TFTP configuration file upload processing.

; error 1: not supported with current hardware

This error indicates that you entered a command that is not supported by the particular the router model you have, or the specified WAN interface module(s). For example, you may have issued the interface sdsl id pvc command, but the SDSL wan interface module in the specified slot is a frame-based SDSL (R7100) interface rather than a cell-based SDSL (R7200) interface.

; error 2: not supported with current configuration

This error indicates that you entered a command that is not compatible with the current configuration of the router. For example, you may have issued a command specific to an ISDN interface in switched mode, but the specified ISDN interface is currently configured for leased mode.

Example:

```
#sh int isdn 1 mode
interface isdn 1 mode idsl-cmn
#int isdn 1 spid 555-1212
; error 2: not supported with current configuration
```

; error 3: not supported

This error indicates that you entered a command in a context in which it was not supported. For example, you may have included the clear command in a text configuration file uploaded via TFTP.

; error 102: can't delete

This error indicates that you attempted to delete an item that doesn't exist.

Example:

```
#show interface ethernet 0 address-serve helper
interface ethernet 0 address-serve helper 10.0.0.1
interface ethernet 0 address-serve helper 20.0.0.1
interface ethernet 0 address-serve helper 30.0.0.1
interface ethernet 0 address-serve helper 40.0.0.1
#no interface ethernet 0 address-serve helper 50.0.0.1
; error 102: can't delete
```

; error 103: incomplete command

This error indicates that you omitted a portion of the command.

Example:

```
#interface ethernet 0 ip
; error 103: incomplete command
```

; error 104: ambiguous

This error indicates that you you didn't enter enough of the text of a keyword such that the keyword as entered was ambiguous.

Example:

```
#sh cp 1 t
; error 104: ambiguous
#sh cp 1 tag
cp 1 tag "Profile 01"
```

; error 106: arp cache is full delete an entry to make room

This error indicates that you attempted to add an entry to the global arp cache when it already contained the maximum number of entries (16).

Index of Commands

D

N

P

R

S

T

U

V

W