Telecommunications Group

3641-80 / 3648-80
Ethernet Routers
Command Line Interface Manual
Section 364-180-C02
Equipment Issue 1
First Printing, April 2006

Revision History

Updates to this document are shown in the tale below.

Revision	Date	Comments
Issue 1.0	April, 2006	

Contents

Introduction to the CLI

1.1. Using CLI and Console Commands

These console commands support both the 3641-80 single port and the 3648-80 routers. There are two types of commands available for use in the router cards: one is CLI commands, the other is Console commands. Users with appropriate access permissions (superuser) can enter console mode from the CLI by entering the "console enable" command and use the console commands. Most of the console commands are the same with the CLI commands. Basically, users can use "help", "help all", "home", "exit" and etc…to search for help and switch the command mode.

The details of each CLI command are described in this manual. While for Console commands, there are only a few commands described in this manual (see appendix) since the Console commands are mainly for customer support debug.

1.2. Help Text for Using the CLI Commands

Within the CLI, the following functions can be used:
* Hitting ? halfway through a word shows all valid completions of that prefix
* Hitting ? after a word shows a list of the words that can follow it
* Hitting TAB halfway through a word completes it, if it is unique
* The UP and DOWN cursor keys move back and forward through the command history
* LEFT and RIGHT cursor keys can be used for line-editing, and CTRL+A and CTRL+E move the cursor to the start and end of the line respectively

Pressing ? at the top-level prompt will display a list of the command groups available. Typing one followed by a space and then hitting ? will show the subcommands within that group, and so on.

Task	Command
List all command groups	**?** `ex. ?`
List all commands under a group	*commnd group* **?** `ex. ethernet ?`

1.3. Using the *source* CLI commands

The *source* <*filename*> command allows you to run a list of predefined commands stored in an existing file. This saves you having to retype lengthy configurations that you will want to use again. Before you can use this command, you need to create a plain text file containing the command list and save it in your ISFS directory. Once you specify the *filename* in the *source* command, the file is located and the commands are executed. For example:

```
--> source //isfs/myconfiguration.txt
Sourcing file '//isfs/myconfiguration.txt'...

--> ethernet clear transports
```

```
--> ethernet add transport eth1 ethernet

--> bridge add interface bridge1

--> bridge attach bridge1 eth1

--> framerelay add transport fr1 fr 171

--> framerelay set transport fr1 encapsulation bridgedether

--> bridge add interface bridge2

--> bridge attach bridge2 fr1

--> ethernet list transports

Ethernet transports:

 ID   |    Name    |    Port
------|------------|------------
   1  | eth1       | ethernet
-------------------------------

--> bridge list interfaces

Bridge Interfaces:

 ID  |   Name     | Filter Type |    Transport
-----|----------|-------------|------------------
   1 | bridge2   | All          | fr1
   2 | bridge1   | All          | eth1
-------------------------------------------------

--> framerelay list transports

Frame Relay Transports:

 ID   |    Name    |    Port    |  DLCI  | Encapsulation
------|------------|------------|--------|----------------------
   1  | fr1        | fr         | 171    | BridgedEther
-------------------------------------------------------------------
```

1.4. Notation Conventions

The notation conventions for the parameter syntax of each CLI command are as follows:

- Parameter values enclosed in < > must be specified.
- Parameters enclosed in [ ] are optional.
- Parameter values are separated by a vertical bar **"|"** only when one of the specified values can be used.
- Parameter values are enclosed in **{ }** when you must use one of the values specified.
- Text in *italics* in a command description indicates commands

WARNING - CLI commands are **case sensitive** when entered

Bridge CLI Commands

1.5. bridge add interface

Syntax

bridge add interface <name>

Description

This command adds a named interface to the bridge.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the interface. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A

Example

prompt> **bridge add interface bridge1**

1.6. bridge attach

Syntax

bridge attach {<name>|<number>} <transport>

Description

This command attaches an existing transport to an existing bridge interface to allow data to be bridged via the transport. Only one transport can be attached to an interface. If you use this command when there is already a transport attached to the interface, the previous transport is replaced by the new one. This command implicitly enables the transport being attached.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing bridge interface. To display interface names, use the *bridge list interfaces*command.	N/A
number	A number that identifies an existing bridge interface. To display interface numbers, use the *bridge list interfaces* command. The number appears in the first column under the heading *ID*.	N/A
transport	A name that identifies an existing transport. To display transport names, use the *<transport type> list transports* command.	N/A

Example

prompt> **bridge attach bridge1 my1483**

1.7. bridge clear interfaces

Syntax

bridge clear interfaces

Description

This command deletes all bridge interfaces that were created using the *bridge add interfac*e command. Any source MAC forwarding rules associated with the interfaces are also deleted by this command.

Example

prompt> **bridge clear interfaces**

1.8. bridge delete interface

Syntax

bridge delete interface {<name>|<number>}

Description

This command deletes a single interface from the bridge configuration. All source MAC forwarding rules associated with the interface that you want to delete are also deleted by this command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing bridge interface. To display interface names, use the *bridge list interfaces* command.	N/A
number	A number that identifies an existing bridge interface. To display interface numbers, use the *bridge list interfaces* command. The number appears in the first column under the heading *ID*.	N/A

Example

prompt> **bridge delete interface 1**

1.9. bridge detach

Syntax

bridge detach <name>

Description

This command detaches the transport that was attached to the bridge interface using the *bridge attach interface* command. All source MAC forwarding rules associated with the interface that you want to detach are deleted by this command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing bridge interface. To display interface names, use the *bridge list interfaces* command.	N/A

Example

prompt> **bridge detach bridge1**

1.10. bridge list interfaces

Syntax

bridge list interfaces

Description

This command lists all bridge interfaces that have been created using the *bridge add interface* command. It displays the following information about bridge interfaces:

- interface ID number
- interface name
- filter type
- name of attached transport (if applicable)

Example

prompt> **bridge list interfaces**

```
Bridge Interfaces:
ID | Name    | Filter Type | Transport
----|-----------|---------------|-------------
1    | bridge1 | All             | eth1
2    | bridge2 | All             | eth2
3    | bridge3 | All             | eth3
---------------------------------------------
```

1.11. bridge set filterage

Syntax

bridge set filterage <filter age>

Description

This command specifies the maximum age of filter table entries for the bridge. The filter age for the bridge is displayed by the *bridge show interface* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
filter age	The filter age is the time (in seconds) after which MAC addresses are removed from the filter table when there has been no activity. The time may be an integer value between 10 and 100,000 seconds.	300 seconds

Example

prompt> **bridge set filterage 2000**

1.12. bridge set interface filtertype

Syntax

bridge set interface {<name>|<number>} filtertype {all|ip|pppoe}

Description

This command specifies the type of ethernet filtering performed by the named bridge interface.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing bridge interface. To display interface names, use the *bridge list interfaces* command.	N/A
number	A number that identifies an existing bridge interface. To display interface numbers, use the *bridge list interfaces* command. The number appears in the first column under the heading *ID*.	N/A
all	Allows all types of ethernet packets through the port.	All
ip	Allows only IP/ARP types of ethernet packets through the port.	
pppoe	Allows only PPPoE types of ethernet packets through the port.	

Example

prompt> **bridge set interface bridge2 filtertype ip**

1.13. bridge set spanning

Syntax

bridge set spanning {enabled|disabled}

Description

This command specifies whether or not the bridge is to implement the spanning tree protocol (STP). The current spanning tree setting is displayed by the *bridge show* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enabled	Allows the bridge to use the spanning tree protocol.	disabled
disabled	Ensures that the bridge acts as a transparent bridge.	

Example

prompt> **bridge set spanning enabled**

1.14. bridge set spanning forwarddelay

* This command is for future feature.

Syntax

bridge set spanning forwarddelay <delay>

Description

This command sets the time that the bridge spends in listening or learning states when the bridge is or is attempting to become the root bridge. The current *forwarddelay* setting is displayed by the *bridge show* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
delay	This can be any value (in seconds) between 4 and 30. BUT it is constrained by the maxage and hellotimes. The maxage, hellotime and forwarddelay times are constrained as follows: 2 x (forwarddelay - 1)≧maxage maxage>2 x (hellotime + 1) For example, the default settings are: 2 x (15 - 1)≧20 20>2 x (2 + 1)	15

Example

prompt> **bridge set spanning forwarddelay 20**

1.15. bridge set spanning hellotime

* This command is for future feature.

Syntax

bridge set spanning hellotime <hellotime>

Description

This command sets the time after which the spanning tree process sends notification of topology changes to the root bridge. This is used when the bridge is or is attempting to become the root bridge. The *hellotime* setting is displayed by the *bridge show* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value

hellotime	This can be any value (in seconds) between 1 and 10. BUT it is constrained by the maximum age and forwarddelay times. The maxage, hellotime and forwarddelay times are constrained. For an example of the constraints, see *Options* for bridge set spanning command.	2

Example

prompt> **bridge set spanning hellotime 10**

1.16. bridge set spanning maxage

* This command is for future feature.

Syntax

bridge set spanning maxage <maxage>

Description

This command sets the maximum age of received spanning tree protocol information before it is discarded. This is used when the bridge is or is attempting to become the root bridge. The *maxage* setting is displayed by the *bridge show* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
maxage	This can be any value (in seconds) between 6 and 40. BUT it is constrained by the hellotime and forwarddelay times. The maxage, hellotime and forwarddelay times are constrained. For an example of the constraints, see *Options* for bridge set spanning command.	20

Example

prompt> **bridge set spanning maxage 30**

1.17. bridge set spanning priority

Syntax

bridge set spanning priority <priority>

Description

This command sets the spanning tree protocol priority. Where two bridges have the same priority, their MAC address is compared and the smaller MAC address is treated as the most significant.
Spanning tree must be enabled before you can use this command. The *priority* setting is displayed by the *bridge show* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
priority	A value that assigns priority to the bridge: the lower the priority number, the more significant the bridge becomes in protocol terms. The priority can be any value (in seconds) between 0 and 65535.	32768

Example

prompt> **bridge set spanning priority 1000**

1.18. bridge show

Syntax

bridge show

Description

This command shows the global configuration settings for the bridge.
The following bridge information is displayed:
• filter age
• spanning tree setting (true or false)
• spanning tree priority value
• spanning tree forward delay time (seconds)
• spanning tree hello time (seconds)
• spanning tree maximum age (seconds)

Example

prompt> **bridge show**
Global bridge configuration:
Filter age: 2000
Spanning bridge configuration:
Spanning: true
Priority: 1000
Forward delay: 20
Hello time: 10
Max. age: 30

1.19. bridge show interface

Syntax

bridge show interface {<name>|<number>}

Description

This command displays the filter type value and portfilter setting of a named bridge interface.
Note - This command **does not** shows the current contents of the bridge's filter table.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing bridge interface. To display interface names, use the *bridge list interfaces* command.	N/A

number	A number that identifies an existing bridge interface. To display interface numbers, use the *bridge list interfaces* command. The number appears in the first column under the heading *ID*.	N/A

Examples

prompt> **bridge show interface bridge1**
Bridge Interface: bridge1
Filter Type: Pppoe

Console Access CLI Commands

1.20. console enable

Syntax

console enable

Description

This command allows you to enter console mode in order to use the console commands. Only superusers can use this command.

Example

prompt> **console enable**
Switching from CLI to console mode - type 'exit' to return

1.21. console process

Syntax

console process <console command>

Description

This command allows you to enter a single *usable* console command without switching to console mode. You cannot enter *blacklisted* console commands using this CLI command. Users with engineer or superuser access can use this command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
console command	A **usable** console command.	N/A

Example

The following *console process* example enters the *usable* console command, *bridge portfilter*:
prompt> **console process bridge portfilter**
portfilter 2 all
portfilter 3 all

DHCP Client CLI commands
This chapter describes the DHCP Client CLI commands.

1.22. dhcpclient add interfaceconfig

Syntax
dhcpclient add interfaceconfig <name> <ipinterface>

Description
This command configures DHCP client parameters for negotiation over an existing IP interface. The client interface can only set the IP configuration if the IP interface has DHCP enabled.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the client interface. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A
ip interface	An IP address or a name that identifies an existing IP interface. The interface must have DHCP enabled. To display interface names, use the *ip list interfaces* command.	N/A

Example
prompt> **dhcpclient add interfaceconfig config1 ip1**

1.23. dhcpclient clear interfaceconfigs

Syntax
dhcpclient clear interfaceconfigs

Description
This command deletes all existing DHCP client interface configurations.

Example
prompt> **dhcpclient clear interfaceconfigs**

1.24. dhcpclient delete interfaceconfig

Syntax
dhcpclient delete interfaceconfig {<name>|<number>}

Description
This command deletes a single DHCP client interface configuration.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A

Example

prompt> **dhcpclient delete interfaceconfig config1**

1.25. dhcpclient interfaceconfig add requested option

Syntax

dhcpclient interfaceconfig {<name>|<number>} add requested option <option>

Description

This command tells the DHCP client to request a specified option from a DHCP server. The requested option *is not* compulsory - if the option is not included in a lease offered by DHCP server, the DHCP client will still accept the offer. Options are detailed in RFC 2132.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
option	A text string that identifies a DHCP server configuration option.	N/A

Example

prompt> **dhcpclient interfaceconfig client1 add requested option irc-server**

1.26. dhcpclient interfaceconfig add required option

Syntax

dhcpclient interfaceconfig {<name>|<number>} add required option <option>

Description

This command tells DHCP client that it requires a specified option from DHCP server. The required option *is* compulsory - if the option is not included in a lease offered by DHCP server, the DHCP client will ignore the offer. Options are detailed in RFC 2132.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
option	A text string that identifies a DHCP server configuration option.	N/A

Example

prompt> **dhcpclient interfaceconfig client1 add required option domain-name**

1.27. dhcpclient interfaceconfig add sent option

Syntax

dhcpclient interfaceconfig {<name>|<number>} add sent option <option> <value>

Description

This command tells the DHCP client to send a value for the given DHCP configuration option to a DHCP server. The DHCP server's response depends on the type of option being sent out.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
option	A text string that identifies a DHCP server configuration option.	N/A
value	The value associated with the option identifier.	N/A

Example

prompt> **dhcpclient interfaceconfig client1 add sent option host-name '"vancouver"'**

This command example tells the DHCP client to send the DHCP hostname option to the DHCP server with the value "vancouver". Note that for options with string-type values associated with them, the option value **must** be in double-quotes (" "). Also, the entire string including the double quotes **must** be inside single quotes (') to ensure that the CLI treats the double quotes literally.

1.28. dhcpclient interfaceconfig clear sent options

Syntax

dhcpclient interfaceconfig {<name>|<number>} clear sent options

Description

This command deletes all options that were previously added to an interfaceconfig using the *dhcpclient interfaceconfig add sent option* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A

Example

prompt> **dhcpclient interfaceconfig client1 clear sent options**

1.29. dhcpclient interfaceconfig clear requested options

Syntax

dhcpclient interfaceconfig {<name>|<number>} clear requested options

Description

This command deletes all options that were previously added to an interfaceconfig using the *dhcpclient interfaceconfig add requested/required option* commands.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A

Example

prompt> **dhcpclient interfaceconfig client1 clear requested options**

1.30. dhcpclient interfaceconfig delete requested option

Syntax

dhcpclient interfaceconfig {<name>|<number>} delete requested option <option number>

Description

This command deletes a single option that was previously added to an interfaceconfig using the *dhcpclient interfaceconfig add requested/required option* commands.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
option number	A number that identifies an option that is requested from the DHCP server by the DHCP client. To display option numbers, use the *dhcpclient interfaceconfig list requested options* command.	N/A

Example

prompt> **dhcpclient interfaceconfig client1 delete requested option 1**

1.31. dhcpclient interfaceconfig delete sent option

Syntax

dhcpclient interfaceconfig {<name>|<number>} delete sent option <option number>

Description

This command deletes a single option that was previously added to an interfaceconfig using the *dhcpclient interfaceconfig add sent option* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
Option number	A number that identifies an option that is sent from the DHCP client to the DHCP server. To display option numbers, use the *dhcpclient interfaceconfig list sent options* command.	N/A

Example

prompt> **dhcpclient interfaceconfig client1 delete sent option 5**

1.32. dhcpclient interfaceconfig list requested options

Syntax

dhcpclient interfaceconfig {<name>|<number>} list requested options

Description

This command lists the options that the DHCP client requests and/or requires from the DHCP server. These options were set using the *dhcpclient interfaceconfig add requested/required option* commands. The following information is displayed:

- Option identification number
- Option identifier (name)
- Requirement status - *true* for options that were added using the *dhcpclient interfaceconfig add required option* command, *false* for options added using the *dhcpclient interfaceconfig add requested option* command.

Options and their values are detailed in RFC2132.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A

Example

prompt> **dhcpclient interfaceconfig client1 list requested options**

```
ID | Identifier          | Is option required?
---|---------------------|--------------------
1  | host-name           | true
2  | domain-name         | false
-------------------------------------------------
```

1.33. dhcpclient interfaceconfig list sent options

Syntax

dhcpclient interfaceconfig {<name>|<number>} list sent options

Description

This command displays a list of the options that the DHCP client sends to the DHCP server. These options were set using the *dhcpclient interfaceconfig add sent option* command. The following information is displayed:

- Option identification number
- Option identifier (name)
- Suggested value

Options and their values are detailed in RFC2132.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A

Example

prompt> **dhcpclient interfaceconfig client1 list sent options**

```
DHCP client options to be sent to server for client1:
ID | Identifier         | Suggested Value
----|--------------------|--------------------
1    | host-name        | vancouver
2    | domain-name     | tailyn
-----------------------------------------------
```

1.34. dhcpclient list interfaceconfigs

Syntax

dhcpclient list interfaceconfigs

Description

This command lists the following information about existing DHCP client interfaces:

- interface identification number
- interface name
- IP interface configured by the client interface
- requested lease time (in seconds)
- client identifier (if set)
- Status of IP address auto-configuration (true or false)

Example

prompt> **dhcpclient list interfaceconfigs**

```
DHCP Client Declarations:
Requested
ID | Name      | Interface | Lease Time | Client ID             | AutoIP
---- |------------|------------|--------------- |---------------------- |--------
1    | client1    | ip1          | 9000           | 00:11:22:33:44:5a | true
-------------------------------------------------------------------------------
```

1.35. dhcpclient set backoff

Syntax

dhcpclient set backoff <backofftime>

Description

This command sets the global maximum time (in seconds) that a DHCP client interface will 'back off' between issuing individual DHCP requests. This prevents many clients trying to configure themselves at the same time, and sending too many requests at once.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
backofftime	The maximum number of seconds that the DHCP client can pause for between unsuccessful DHCP negotiations.	120

Example

prompt> **dhcpclient set backoff 200**

1.36. dhcpclient set interfaceconfig autoip enabled|disabled

dhcpclient set interfaceconfig {<name>|<number>} autoip {enabled | disabled}

Description

This command enables/disables IP address auto-configuration (Auto-IP). Auto-IP automatically configures an IP address when a DHCP client fails to contact a DHCP server and cannot obtain a lease. An IP address on the 169.254 subnet is automatically created, and ARP requests are issued for the suggested IP address. The address is abandoned if it already exists on the network or if any other host on the network issues an ARP probe for that IP address. Once an IP address has been automatically configured, the DHCP client continues to check whether or not it can contact a DHCP server. If the client can contact a DHCP server and obtain a legitimate lease, the legitimate lease will supersede the auto-configured IP address. Even if you have enabled Auto-IP using this command, you will not be able to use IP address auto-configuration if a DHCP server on the same network does not allow it. See the *dhcpserver subnet add option* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
enabled	Enables Auto-IP on a specified dhcp client.	enabled
disabled	Disables Auto-IP on a specified dhcp client.	

Example

prompt> **dhcpclient set interfaceconfig mycfg autoip enabled**

1.37. dhcpclient set interfaceconfig clientid

Syntax

dhcpclient set interfaceconfig {<name>|<number>} clientid <clientid>

Description

This command sets a unique client identifier that DHCP server uses to identify the client.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
client id	A unique identifier that DHCP server can use to identify the client. For Microsoft DHCP servers, the client ID should be the MAC address of the card that DHCP is running on. For other DHCP servers, the client ID can be a MAC address or a text string such as the hostname.	N/A

Example

prompt> **dhcpclient set interfaceconfig client1 clientid 00:11.22.33.44.5a**

1.38. dhcpclient set interfaceconfig defaultroute enabled|disabled

Syntax

dhcpclient set interfaceconfig {<name>|<number>} defaultroute {enabled|disabled}

Description

This command enables/disables whether DHCP client makes use of default gateway information received from a DHCP server. If no DHCP interfaceconfigs have been added to the system, by default DHCP client will use default gateway information received from DHCP server.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
enabled	DHCP client uses default gateway information it receives from DHCP server.	enabled
disabled	DHCP client does not use default gateway information it receives from DHCP server.	

Example

prompt> **dhcpclient set interfaceconfig client1 defaultroute disabled**

1.39. dhcpclient set interfaceconfig dhcpinform enabled|disabled

Syntax

dhcpclient set interfaceconfig {<name>|<number>} dhcpinform {enabled|disabled}

Description

This command enables/disables whether DHCP client uses the *dhcpinform* message type. This DHCP message type is used whenever a client has obtained an IP address or subnet mask (for example, the address has been manually configured or obtained through PPP/IPCP), but wishes to obtain extra configuration parameters (such as DNS servers or default gateway) from a DHCP server.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
enabled	Enables the dhcpinform message type. IP address and subnet mask will not be negotiated if this mode is selected.	disabled
disabled	Disables the dhcpinform message type.	

Example

prompt> **dhcpclient set interfaceconfig client1 dhcpinform disabled**

1.40. dhcpclient set interfaceconfig dhcpserverpoolsize

Syntax

dhcpclient set interfaceconfig {<name>|<number>} dhcpserverpoolsize <pool size>

Description

This command tells DHCP client to configure a DHCP server on the LAN if the given address pool size is set to a number greater than 0. The LAN DHCP server is configured using parameters received by a DHCP client interface on the WAN. Information such as DNS server addresses can then be distributed to LAN clients. The new DHCP server gives out the default gateway address as its LAN IP address.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
pool size	The number of DHCP client addresses in a pool. The first address in the pool is the address immediately after the LAN DHCP address. For example, if the LAN DHCP address is 192.168.102.3, the first address in the pool will be 192.168.102.4.	N/A

Example

prompt> **dhcpclient set interfaceconfig client1 dhcpserverpoolsize 5**

1.41. dhcpclient set interfaceconfig dhcpserverinterface

Syntax

dhcpclient set interfaceconfig {<name>|<number>} dhcpserverinterface <interface name>

Description

This command allows the user to specify an existing IP interface on which the automatically configured DHCP server can be created. If the interface name does not correspond with an existing IP interface, or no interface name is given, the DHCP server will be placed on the first LAN interface that it finds.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
Interface name	The name that identifies an existing IP interface. To display IP interface names, use the *ip list interfaces* command.	N/A

Example

prompt> **dhcpclient set interfaceconfig client1 dhcpserverinterface ip2**

1.42. dhcpclient set interfaceconfig givednstoclient enabled|disabled

Syntax

dhcpclient set interfaceconfig {<name>|<number>} givednstoclient {enabled|disabled}

Description

This command enables/disables whether DHCP client passes received DNS server addresses to DNS client. If no DHCP interfaceconfigs have been added to the system, by default DHCP client will not pass DNS server addresses to DNS client.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
enabled	DHCP client passes DNS server addresses to DNS client.	disabled
disabled	DHCP client does not pass DNS server addresses to DNS client.	

Example

prompt> **dhcpclient set interfaceconfig client1 givednstoclient disabled**

1.43. dhcpclient set interfaceconfig givednstorelay enabled|disabled

Syntax

dhcpclient set interfaceconfig {<name>|<number>} givednstorelay {enabled|disabled}

Description

This command enables/disables whether DHCP client passes received DNS server addresses to DNS relay. If no DHCP interfaceconfigs have been added to the system, by default DHCP client will pass DNS server addresses to DNS relay.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
enabled	DHCP client passes DNS server addresses to DNS relay.	enabled
disabled	DHCP client does not pass DNS server addresses to DNS relay	

Example

prompt> **dhcpclient set interfaceconfig client1 givednstorelay disabled**

1.44. dhcpclient set interfaceconfig interface

Syntax

dhcpclient set interfaceconfig {<name>|<number>} interface <ipinterface>

Description

This command sets the IP interface that will have its configuration set by the DHCP client interface. The client interface can only set the IP configuration if the IP interface has DHCP enabled, using the *ip set interface dhcp* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
ipinterface	A name that identifies an existing IP interface. The interface must have DHCP enabled. To display interface names, use the *ip list interfaces* command.	N/A

Example

prompt> **dhcpclient set interfaceconfig client1 interface ip2**

1.45. dhcpclient set interfaceconfig noclientid

Syntax

dhcpclient set interfaceconfig {<name>|<number>} noclientid

Description

This command deletes a client identifier from a DHCP client. The DHCP server must have 'allowunknownclients' enabled in order to work with DHCP clients that are not specifically named in DHCP server configuration or its lease database.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A

Example

prompt> **dhcpclient set interfaceconfig client1 noclientid**

1.46. dhcpclient set interfaceconfig requestedleasetime

Syntax

dhcpclient set interfaceconfig {<name>|<number>} requestedleasetime <requestedleasetime>

Description

The DHCP client requests a specific lease time from the DHCP server for the allocated IP addresses. This command determines the length of lease time requested. The DHCP server will 'cap' a requested lease time if it is too large.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing DHCP client interface. To display client interface names, use the *dhcpclient list interfaceconfigs* command.	N/A
Number	A number that identifies an existing DHCP client interface. To display client interface numbers, use the *dhcpclient list interfaceconfigs* command.	N/A
Requested lease time	The lease time (in seconds) that a DHCP client requests from the DHCP server.	86400

Example

prompt> **dhcpclient set interfaceconfig client1 requestedleasetime 70000**

1.47. dhcpclient set reboot

Syntax

dhcpclient set reboot <reboottime>

Description

When the DHCP client is restarted, it tries to reacquire the last address that it had. This command sets the time between the client trying to reacquire its last address and giving up then trying to discover a new

address.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
reboottime	The time (in seconds) after a client tries to reacquire the last IP address it had and before the client gives up then tries to discover a new address.	10

Example

prompt> **dhcpclient set reboot 5**

1.48. dhcpclient set retry

Syntax

dhcpclient set retry <retrytime>

Description

This command sets the time that must pass after the client has determined that no DHCP server is present before it tries again to contact a DHCP server.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
retrytime	The time (in seconds) that must pass after the client has determined that no DHCP server is present before it tries again to contact a DHCP server.	300

Example

prompt> **dhcpclient set retry 150**

1.49. dhcpclient show

Syntax

dhcpclient show

Description

This command displays the following global configuration information about DHCP client:

- reboot time
- retry time
- maximum backoff time

Example

prompt> **dhcpclient show**
Global DHCP Client Configuration:
Reboot time: 10
Retry time: 300
Max. backoff time: 120

1.50. dhcpclient show interfaceconfigs

Syntax

dhcpclient show interfaceconfigs <name>

Description

This command shows the specific information about existing DHCP client interfaces:

- interface name

- requested lease time (in seconds)
- client identifier(if set)
- Status of IP address auto-configuration (true or false)
- Status of give DNS info to DNS relay (true or false)
- Status of give DNS info to DNS client (true or false)
- Auto DHCP server pool size
- Auto DHCP server interface

Example

prompt> **dhcpclient show interfaceconfigs client1**

DHCP Client Interface Declaration: client1

IP Interface: ip1
Requested lease time: 86000 seconds
Client identifier:
Can use Auto IP: true
Give DNS info to DNS relay: true
Give DNS info to DNS client: false
Auto DHCP server pool size: 0
Auto DHCP server interface:

1.51. dhcpclient update

Syntax

dhcpclient update

Description

This command updates the DHCP client configuration. Changes made to the client configuration are not updated until this command has been entered.

Example

prompt> **dhcpclient update**
dhcpclient: Reset request acknowledged. Reset imminent.

DHCP Relay CLI commands
This chapter describes the DHCP Relay CLI commands.

1.52. dhcprelay add server

Syntax
dhcprelay add server <ipaddress>

Description
This command adds the IP address of a DHCP server subnet to DHCP relay's list of server IP addresses. Records of new IP addresses added are not updated until the *dhcprelay update* command has been entered.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
ipaddress	The IP address of a DHCP server that DHCP relay can use. The IP address is displayed in the following format: 192.168.102.3	N/A

Example
prompt> **dhcprelay add server 239.252.197.0**

1.53. dhcprelay clear servers

Syntax
dhcprelay clear servers

Description
This command deletes all DHCP server IP addresses stored in DHCP relay's list of server IP addresses.

Example
prompt> **dhcprelay clear servers**

1.54. dhcprelay delete server

Syntax
dhcprelay delete server <number>

Description
This command deletes a single DHCP server address stored in DHCP relay's list of server IP addresses.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	A number that identifies the DHCP server in the DHCP relay list. To display server numbers, use the *dhcprelay list servers* command.	N/A

Example
prompt> **dhcprelay delete server 3**

1.55. dhcprelay enable|disable

Syntax
dhcprelay {enable|disable}

Description

This command enables/disables DHCP relay. You must have DHCP relay enabled in order to carry out any DHCP relay configuration. If you try configuring DHCP relay before you've entered the *dhcprelay enable* command, the CLI issues a warning message. You **cannot** have DHCP relay and DHCP server enabled at the same time. If you try to configure DHCP relay when DHCP server is enabled, the CLI issues a warning message.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enable	Enables configuration of DHCP relay.	enable
disable	Disables configuration of DHCP relay.	

Example

prompt> **dhcprelay enable**

1.56. dhcprelay list servers

Syntax

dhcprelay list servers

Description

This command displays the DHCP relay's list of DHCP server IP addresses with their identification numbers.

Example

```
prompt> dhcprelay list servers
DHCP Servers:
ID |    IP Address
----|----------------------
1    |    192.168.102.3
2    |    239.252.197.0
---------------------------
```

1.57. dhcprelay show

Syntax

dhcprelay show

Description

This command tells you whether DHCP relay is enabled or disabled.

Example

prompt> **dhcprelay show server**
Global DHCP Relay Configuration:
Status: ENABLED

1.58. dhcprelay update

Syntax

dhcprelay update

Description

This command updates the DHCP relay configuration. Changes made to the relay configuration will not take effect until this command has been entered.

Example

prompt> **dhcprelay update**
dhcprelay: Reset request acknowledged. Reset imminent.

DHCP Server CLI commands
This chapter describes the DHCP Server CLI commands.

1.59. dhcpserver add subnet

Syntax
dhcpserver add subnet <name> <ipaddress> <netmask> [<startaddr> <endaddr>]

Description
This command creates a subnet that stores a pool of IP addresses. The DHCP server can allocate IP addresses from this pool to clients on request.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the subnet. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A
ipaddress	The IP address of the subnet, displayed in the following format: 192.168.102.3	N/A
netmask	The netmask address of the subnet, for example: 255.255.255.0	N/A
startaddr	The first IP address in the pool of addresses. The IP address is displayed in the following format: 192.168.102.3	N/A
endaddr	The last IP address in the pool of addresses. The IP address is displayed in the following format: 192.168.102.3	N/A

Example
prompt>**dhcpserver add subnet sub1 239.252.197.0 255.255.255.0 239.252.197.10 239.252.197.107**
Note: The maximum number of DHCP IP addresses supported by the system is 128.

1.60. dhcpserver clear subnets

Syntax
dhcpserver clear subnets

Description
This command deletes all DHCP server subnets that were created using the *dhcpserver add subnet* commands.

Example
prompt> **dhcpserver clear subnets**

1.61. dhcpserver delete subnet

Syntax
dhcpserver delete subnet {<name>|<number>}

Description
This command deletes a single DHCP server subnet. The pool of IP addresses in the subnet are also deleted.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A

Example

prompt> **dhcpserver delete subnet sub1**

1.62. dhcpserver enable|disable

Syntax

dhcpserver {enable|disable}

Description

This command enables/disables the DHCP server. You must have the DHCP server enabled in order to carry out any DHCP server configuration. If you try configuring DHCP server when *dhcpserver disable* is set, the CLI issues a warning message. You **cannot** have DHCP server and DHCP relay enabled at the same time. If you try to configure DHCP server when DHCP relay is enabled, the CLI issues a warning message.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enable	Enables configuration of the DHCP server.	enable
disable	Disables configuration of the DHCP server.	

Example

prompt> **dhcpserver enable**

1.63. dhcpserver list options

Syntax

dhcpserver list options

Description

This command lists the option data types available for DHCP server. These options are detailed in RFC2132. You can configure the DHCP server using any of the options listed.

Example

prompt> **dhcpserver list options**
subnet-mask
time-offset
routers
time-servers
ien116-name-servers
domain-name-servers
log-servers
cookie-servers
lpr-servers

impress-servers
resource-location-servers
host-name
boot-size
merit-dump
domain-name
swap-server
root-path
extensions-path
ip-forwarding
non-local-source-routing
policy-filter
max-dgram-reassembly
default-ip-ttl
path-mtu-aging-timeout
path-mtu-plateau-table
interface-mtu
all-subnets-local
broadcast-address
perform-mask-discovery
mask-supplier
router-discovery
router-solicitation-address
static-routes
trailer-encapsulation
arp-cache-timeout
ieee802-3-encapsulation
default-tcp-ttl
tcp-keepalive-interval
tcp-keepalive-garbage
nis-domain
nis-servers
ntp-servers
vendor-encapsulated-options
netbios-name-servers
netbios-dd-server
netbios-node-type
netbios-scope
font-servers
x-display-manager
dhcp-requested-address
dhcp-lease-time
dhcp-option-overload
dhcp-message-type
dhcp-server-identifier
dhcp-parameter-request-list
dhcp-message
dhcp-max-message-size
dhcp-renewal-time
dhcp-rebinding-time

dhcp-class-identifier
dhcp-client-identifier
option-62
option-63
nisplus-domain
nisplus-servers
tftp-server-name
bootfile-name
mobile-ip-home-agent
smtp-server
pop-server
nntp-server
www-server
finger-server
irc-server
streettalk-server
streettalk-directory-assistance-server
user-class
option-78
option-79
option-80
option-81
option-82
option-83
option-84
nds-servers
nds-tree-name
nds-context
option-88
option-89
...(more options down to)
option-115
auto-configure
option-117
...(more options down to)
option-254
option-end

1.64. dhcpserver list subnets

Syntax

dhcpserver list subnets

Description

This command lists the following information about existing DHCP server subnets:

- subnet number
- subnet name
- subnet IP address
- subnet netmask address
- default lease time (in seconds)
- maximum lease time (in seconds)
- whether the host is a DNS server (true or false)

Example

prompt> **dhcpserver list subnets**
DHCP Server subnets:
Default Max Host is
ID	IP Address	Netmask	Lease time	Lease time	DNS svr
1	192.168.102.0	255.255.255.0	43200	86400	false

1.65. dhcpserver set allowunknownclients

Syntax

dhcpserver set allowunknownclients {enabled|disabled}

Description

This command enables/disables the dynamic assignment of addresses to unknown clients.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enabled	Allows IP addresses to be dynamically assigned to unknown clients.	enabled
disabled	Does not allow IP addresses to be dynamically assigned to unknown clients.	

Example

prompt> **dhcpserver set allowunknownclients disabled**

1.66. dhcpserver set bootp

Syntax

dhcpserver set bootp {enabled|disabled}

Description

This command determines whether or not DHCP server can respond to BOOTP requests.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enabled	DHCP server responds to BOOTP queries.	enabled
disabled	DHCP server does not respond to BOOTP queries.	

Example

prompt> **dhcpserver set bootp disabled**

1.67. dhcpserver set defaultleasetime

Syntax

dhcpserver set defaultleasetime <defaultleasetime>

Description

This command sets the global default lease time for DHCP server.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value

defaultleasetime	The default time (in seconds) that is assigned to a lease if the client requesting the lease does not ask for a specific expiry time.	43200

Example

prompt> **dhcpserver set defaultleasetime 50000**

1.68. dhcpserver set maxleasetime

Syntax

dhcpserver set maxleasetime <maxleasetime>

Description

This command sets the global maximum lease time for DHCP server.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
maxleasetime	The maximum time (in seconds) that is assigned to a lease if the client requesting the lease does not ask for a specific expiry time.	86400

Example

prompt> **dhcpserver set maxleasetime 90000**

1.69. dhcpserver set subnet defaultleasetime

Syntax

dhcpserver set subnet {<name>|<number>} defaultleasetime <defaultleasetime>

Description

This command sets the default lease time for an existing subnet. This command setting overrides the global default lease time setting for this particular subnet.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A
defaultleasetime	The default time (in seconds) that a subnet assigns to a lease if the client requesting the lease does not ask for a specific expiry time.	43200

Example

prompt> **dhcpserver set subnet sub1 defaultleasetime 30000**

1.70. dhcpserver set subnet hostisdefaultgateway

Syntax

dhcpserver set subnet <{<name>|<number>} hostisdefaultgateway {enabled | disabled}

Description

This command tells the DHCP server to give out its own host IP address as the default gateway address.

This is useful when combined with DNS Relay.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A
enabled	Allows DHCP server to give out its own host IP address as the default gateway address.	disabled
disabled	Disallows DHCP server from giving out its own host IP address as the default gateway address.	

Example

prompt> **dhcpserver set subnet sub1 hostisdefaultgateway enabled**

1.71. dhcpserver set subnet hostisdnsserver

Syntax

dhcpserver set subnet {<name>|<number>} hostisdnsserver {enabled | disabled}

Description

This command tells the DHCP server to give out its own host IP address as the DNS server address. This is useful when combined with DNS Relay.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A
enabled	Allows DHCP server to give out its own host IP address as the DNS server address.	disabled
disabled	Disallows DHCP server from giving out its own host IP address as the DNS server address.	

Example

prompt> **dhcpserver set subnet sub1 hostisdnsserver enabled**

1.72. dhcpserver set subnet maxleasetime

Syntax

dhcpserver set subnet {<name>|<number>} maxleasetime <maxleasetime>

Description

This command sets the maximum lease time for an existing subnet. This command setting overrides the global maximum lease time setting for this particular subnet.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A
maxleasetime	The maximum time (in seconds) that a subnet assigns to a lease if the client requesting the lease does not ask for a specific expiry time.	86400

Example

prompt> **dhcpserver set subnet sub1 maxleasetime 70000**

1.73. dhcpserver set subnet subnet

Syntax

dhcpserver set subnet {<name>|<number>} subnet <ip address> <netmask>

Description

This command allows you to change the IP address and netmask used by an existing DHCP server subnet.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A
ip address	The new IP address for the subnet, displayed in the following format: 192.168.102.3	N/A
netmask	The new netmask address for the subnet, for example: 255.255.255.0	N/A

Example

prompt> **dhcpserver set subnet sub1 subnet 239.252.197.0 255.255.255.0**

1.74. dhcpserver show

Syntax

dhcpserver show

Description

This command displays the following global configuration information about the DHCP server:

- status of the server (enabled/disabled)
- global default lease time
- global maximum lease time
- bootp requests setting (enable/disable)
- allow unknown clients setting (enable/disable)

Example

prompt> **dhcpserver show**
Global DHCP Server Configuration:
Status: ENABLED
Default lease time: 43200 seconds
Max. lease time: 86400 seconds
Allow BOOTP requests: true
Allow unknown clients: true

1.75. dhcpserver show subnet

Syntax

dhcpserver show subnet {<name>|<number>}

Description

This command displays the following information about a subnet:
• subnet name
• subnet IP address
• subnet netmask
• subnet maximum lease time
• subnet default lease time

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A

Example

prompt> **dhcpserver show subnet sub1**
DHCP Server Subnet: sub1
Subnet: 192.168.103.0
Netmask: 255.255.255.0
Max. lease time: 70000 seconds
Default lease time: 30000 seconds

1.76. dhcpserver subnet add iprange

Syntax

dhcpserver subnet {<name>|<number>} add iprange <startaddr> <endaddr>

Description

This command adds a pool of IP addresses to an existing subnet. DHCP server can allocate IP addresses from this pool to clients on request.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets*	N/A

	command.	
number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A
startaddr	The first IP address in the pool of addresses. The IP address is displayed in the following format: 192.168.102.3	N/A
endaddr	The last IP address in the pool of addresses. The IP address is displayed in the following format: 192.168.102.3	N/A

Example

prompt> **dhcpserver subnet sub1 add iprange 239.252.197.0 239.252.197.107**

1.77. dhcpserver subnet add option

Syntax

dhcpserver subnet {<name>|<number>} add option <identifier> <value>

Description

This command allows you to configure the DHCP server using the options detailed in RFC2132. To display a list of available options, use the command *dhcpserver list options*. The heading of each option in the list contains the option identifier and the required value (in italics) for that specific option. The following is an extract from the option list: option auto-configure *flag*; This option, based on RFC2563, controls whether or not the auto configuration of IP address is to be allowed for clients on this subnet. It only applies in cases where the DHCP server is unwilling or unable to supply an IP address lease. In this case, if this option is set to 1, then the DHCP server will not intervene to prevent clients from using auto-configuration to determine an IP address. If this option is set to 0, the use of IP address auto-configuration on the network will be explicitly forbidden by the DHCP server. If this option is not explicitly configured, then it will be assumed that auto-configuration is allowed on the network.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A
identifier	A text string that identifies a DHCP server configuration option.	N/A
value	The value associated with the option identifier.	N/A

Example

prompt> **dhcpserver subnet sub1 add option auto-configure 1**

1.78. dhcpserver subnet clear ipranges

Syntax

dhcpserver subnet {<name>|<number>} clear ipranges

Description

This command deletes all of the IP ranges set for an existing subnet.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
number	A number that identifies an existing subnet. To display subnet numbers, usethe *dhcpserver list subnets* command.	N/A

Example

prompt> **dhcpserver subnet sub1 clear ipranges**

1.79. dhcpserver subnet clear options

Syntax

dhcpserver subnet {<name>|<number>} clear options

Description

This command deletes the options set for an existing subnet.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A

Example

prompt> **dhcpserver subnet sub1 clear options**

1.80. dhcpserver subnet delete iprange

Syntax

dhcpserver subnet {<name>|<number>} delete iprange <range-id>

Description

This command deletes a single IP range from an existing subnet.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A
range-id	A number that identifies an IP range. To list the existing range-ids for a subnet, use the *dhcpserver*	N/A

	subnet list ipranges command.	

Example

prompt> **dhcpserver subnet sub1 delete iprange 1**

1.81. dhcpserver subnet delete option

Syntax

dhcpserver subnet {<name>|<number>} delete option <option number>

Description

This command deletes a single option that was created using the *dhcpserver subnet add option* command.Once deleted, the option will no longer be given out by the DHCP server.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A
Option number	A number that identifies an existing option. To list all existing options, use the *dhcpserver subnet list options* command.	N/A

Example

prompt> **dhcpserver subnet sub1 delete option 2**

1.82. dhcpserver subnet list ipranges

Syntax

dhcpserver subnet {<name>|<number>} list ipranges

Description

This command lists the IP range(s) for an existing subnet that has been added using the *dhcpserver add subnet* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A

Example

prompt> **dhcpserver subnet sub1 list ipranges**

IP Ranges for subnet: sub1

ID | Start Address | End Address

----|---------------------|-----------------------

1 | 192.168.102.0 | 192.168.102.100

2 | 192.168.102.200 | 192.168.102.300

1.83. dhcpserver subnet list options

Syntax

dhcpserver subnet {<name>|<number>} list options

Description

This command lists the options for an existing subnet that has been added using the *dhcpserver add subnet* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
Name	A name that identifies an existing subnet. To display subnet names, use the *dhcpserver list subnets* command.	N/A
Number	A number that identifies an existing subnet. To display subnet numbers, use the *dhcpserver list subnets* command.	N/A

Example

prompt> **dhcpserver subnet sub1 list options**
Options for subnet: sub1
ID	Identifier	Value
1	ip-forwarding	false
2	subnet-mask	255.255.255.0

1.84. dhcpserver update

Syntax

dhcpserver update

Description

This command updates the DHCP server configuration. Changes made to the server configuration will not take effect until this command has been entered.

Example

prompt> **dhcpserver update**
dhcpserver: Reset request acknowledged. Reset imminent.

DNS Client CLI commands
This chapter describes the DNS Client CLI commands.

1.85. dnsclient add searchdomain

Syntax
dnsclient add searchdomain <searchstring>

Description
This command creates a domain search list. The DNS client uses this list when a user asks for the IP address list for an incomplete domain name. The search string specified replaces any previous search strings added previously using this command.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
searchstring	A search string used to find the IP address for an incomplete domain name. You can have a maximum of 6 incomplete domain names in the search string.	N/A

Example
prompt> **dnsclient add searchdomain tailyn.com.tw**

1.86. dnsclient add server

Syntax
dnsclient add server <ipaddress>

Description
This command adds a server IP address to the server list. This enables you to retrieve a domain name for a given IP address.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
ipaddress	The IP address of the server that has an unknown domain name. You can add a maximum of 3 addresses to the server list. The IP address is displayed in the following format: 192.168.102.3	N/A

Example
prompt> **dnsclient add server 192.168.219.196**

1.87. dnsclient clear searchdomains

Syntax
dnsclient clear searchdomains

Description
This command deletes all domain names from the domain search list.

Example
prompt> **dnsclient clear searchdomains**

1.88. dnsclient clear servers

Syntax

dnsclient clear servers

Description

This command deletes all the server IP addresses to the server list.

Example

prompt> **dnsclient clear servers**

1.89. dnsclient delete searchdomain

Syntax

dnsclient delete searchdomain <searchstring>

Description

This command deletes a single domain name from the domain search list.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
searchstring	A number that identifies a search string used to find the IP address for an incomplete domain name. To list domain search strings, use the *dnsclient list searchdomains* command.	N/A

Example

prompt> **dnsclient delete searchdomain 1**

1.90. dnsclient delete server

Syntax

dnsclient delete server <number>

Description

This command deletes a single server IP addresses from the server list.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	The server number that identifies an IP address of the server that has an unknown domain name. To display server numbers, use the *dnsclient list servers* command.	N/A

Example

prompt> **dnsclient delete server 1**

1.91. dnsclient list searchdomains

Syntax

dnsclient list searchdomains

Description

This command lists the domain search strings that you have added to DNS client using the *dnsclient add searchdomain* command. DNS client uses this list when a user asks for the IP address list for an incomplete domain name.

Example

```
prompt> dnsclient list searchdomains
ID | Domain
-----|--------------------
1 | tailyn.com.tw
--------------------------
```

1.92. dnsclient list servers

Syntax

dnsclient list servers

Description

This command lists the server IP addresses that you have added to DNS client using the *dnsclient add server* command. DNS client uses this list to retrieve a domain name for a given IP address.

Example

```
prompt> dnsclient list servers
DNS Client Servers:
ID | IP Address
----|-----------------
1    | 192.168.100.7
2    | 192.168.100.1
-----------------------
```

DNS Relay CLI commands
This chapter describes the DNS (Domain NameServer) Relay CLI commands.

1.93. dnsrelay add server

Syntax
dnsrelay add server <ip-address>

Description
This command adds the IP address of a DNS server to DNS relay's list of server IP addresses. The relay can store a maximum of 10 DNS server addresses.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
ip-address	The IP address of a DNS server that DNS relay can use. The IP address is displayed in the following format: 192.168.102.3	0.0.0.0

Example
prompt> **dnsrelay add server 239.252.197.0**
DNS server set to 0.0.0.0
DNS server set to 239.252.197.0

1.94. dnsrelay clear servers

Syntax
dnsrelay clear servers

Description
This command deletes all DNS server IP addresses stored in DNS relay's list of server IP addresses.

Example
prompt> **dnsrelay clear servers**

1.95. dnsrelay delete server

Syntax
dnsrelay delete server <id-number>

Description
This command deletes a single DNS server address stored in DNS relay's list of server IP addresses.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
ID number	A number that identifies the DNS server in the DNS relay list. To display server numbers, use the *dnsrelay list servers* command.	N/A

Example
prompt> **dnsrelay delete server 3**

1.96. dnsrelay list servers

Syntax

dnsrelay list servers

Description

This command displays the DNS relay's list of DNS server IP addresses with their identification numbers.

Example

prompt> **dnsrelay list servers**
DNS Relay Servers:
ID	IP Address
1	239.252.197.0

Ethernet CLI commands
This chapter describes the Ethernet transport CLI commands.

1.97. ethernet add transport

Syntax

ethernet add transport <name> [<port>]

Description

This command adds a named ethernet transport and allows you to specify which port it will use to transport ethernet data. The ports are defined in the *initbun* file for each type of ATMOS product. For example, for an eth-gateway product, the ports are defined in *atmos/products/eth-gateway/flashfs/initbun*.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the transport. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A
port	The port is used to transport ethernet data. You cannot use the same port for more than one ethernet transport at a time.	Ethernet

Example

prompt> **ethernet add transport eth1 ethernet**

1.98. ethernet clear transports

Syntax

ethernet clear transports

Description

This command deletes all ethernet transports that were created using the *ethernet add transport* command.

Example

prompt> **ethernet clear transports**

1.99. ethernet delete transport

Syntax

ethernet delete transport {<name>|<number>}

Description

This command deletes a single ethernet transport.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing Ethernet transport. To display transport names, use the *ethernet list transports* command.	N/A

number	A number that identifies an existing Ethernet transport. To display transport numbers, use the *ethernet list transports* command.	N/A

Example

prompt> **ethernet delete transport eth1**

1.100. ethernet list ports

Syntax

ethernet list ports

Description

This command lists the valid ports that can be used to transport Ethernet data. The ports are defined in the *initbun* file for each type of ATMOS product. For example, for an eth-gateway product, the ports are defined in *atmos/products/eth-gateway/flashfs/initbun*.

Example

prompt> **ethernet list ports**
Valid ethernet port names:
ethernet
hdlc

1.101. ethernet list transports

Syntax

ethernet list transports

Description

This command lists all ethernet transports that have been created using the *ethernet add transport* command. It displays the transport identification number and name, and the name of the port that it uses to transport ethernet data.

Example

prompt> **ethernet list transports**
Ethernet transports:

```
ID | Name      | Port
----|-----------|-----------
1   | eth2      | hdlc
2   | eth1      | ethernet
```

1.102. ethernet set transport port

Syntax

ethernet set transport {<name>|<number>} port <port>

Description

This command sets the port that an existing Ethernet transport uses to transport ethernet data.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value

name	A name that identifies an existing Ethernet transport. To display transport names, use the *ethernet list transports* command.	N/A
number	A number that identifies an existing Ethernet transport. To display transport numbers, use the *ethernet list transports* command.	N/A
port	The port is used to transport ethernet data. You cannot use the same port for more than one ethernet transport at a time.	Ethernet

Example

prompt> **ethernet set transport eth1 port hdlc**

1.103. ethernet show transport

Syntax

ethernet show transport {<name>|<number>}

Description

This command displays the name and port used by an existing Ethernet transport.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing Ethernet transport. To display transport names, use the *ethernet list transports* command.	N/A
number	A number that identifies an existing Ethernet transport. To display transport numbers, use the *ethernet list transports* command.	N/A

Example

prompt> **ethernet show transport eth1**
Ethernet transport: eth1
Description: Default LAN port
Port: Ethernet

Firewall CLI commands

This chapter describes the stateful Firewall CLI commands.

1.104. firewall add policy

Syntax

firewall add policy <name> {external-internal|externaldmz| dmz-internal}
[{allowonly-val}|{blockonly-val}]

Description

This command creates a policy between two interface types. There are three types of policy that you can add to the firewall:

- a policy between the external interface and the internal interface
- a policy between the external interface and the DMZ interface
- a policy between the DMZ interface and the internal interface

A policy is the collective term for the rules that apply to incoming and outgoing traffic between two interface types. Once you have created a policy using the *firewall add policy* command, you can create rules for it using the *firewall add portfilter* command and the *firewall add validator* commands. The *firewall add validator* command allows you to block/allow traffic based on the source and/or destination IP addresses and masks. The *firewall add policy* command controls whether traffic is blocked/allowed for *all* of the validators that belong to a policy. There are two options:

- **allow only** traffic to and/or from the IP address(es) set in the *firewall add validator* command. All other traffic is **blocked** by the Firewall.
- **block only** traffic to and/or from the IP address(es) set in the *firewall add validator* command. All other traffic is **allowed** through the Firewall.

You can set a Firewall security level that contains default policies using the *firewall set securitylevel* command. You can then customize the Firewall by adding your own portfilters and validators.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the policy. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A
externalinternal	A connection between the external network interface and the internal network interface.	N/A
externaldmz	A connection between the external network interface and the de-militarized zone (DMZ).	
dmz-interna	A connection between the de-militarized zone (DMZ) and the internal network interface.	
allowonlyval	Allows *only* traffic to and/or from the IP address(es) set in the *firewall add validator* command. All other traffic is blocked.	N/A
blockonlyval	Blocks *only* traffic to and/or from the IP address(es) set in the *firewall add validator* command. All other traffic is allowed.	N/A

Example

prompt> **firewall add policy ext-dmz external-dmz blockonly-val**

1.105. firewall add portfilter

Syntax

firewall add portfilter <name> <policyname> {protocol<number>} {inbound|outbound|both}
firewall add portfilter <name> <policyname> {tcp|udp}<startport> <endport> {inbound|outbound|both}
firewall add portfilter <name> <policyname>{icmp|smtp|http|ftp|telnet} {inbound|outbound|both}

Description

This command adds a portfilter to an existing firewall policy. Portfilters are individual rules that determine what kind of traffic can pass between the two interfaces specified in the *firewall add policy* command. There are three ways that you can add a portfilter depending on the type of protocol that you want to feature in the portfilter:

- specify the number of a non-TCP or non-UDP protocol (for more information, see *http://www.ietf.org/rfc/rfc1700.txt*)
- specify TCP or UDP protocol, together with an application's start/end port numbers
- specify one of the listed protocols, applications or services. These are provided by the Firewall as popular examples that you can use. You do not need to specify the portnumber - the Firewall does this for you.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the portfilter. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A
policyname	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A
(protocol) number	The number of a non-TCP or non-UDP protocol. Protocol numbers can be found at *http://www.ietf.org/rfc/rfc1700.txt*.	N/A
startport	The start of the port range for a TCP or UDP protocol.	N/A
endport	The end of the port range for a TCP or UDP protocol.	N/A
inbound	Allows transport of packets of the specified protocol, application or service from an outside interface to an inside interface. Outbound transport of the packets is not allowed.	N/A
outbound	Allows transport of packets of the specified protocol, application or service from an inside interface to an outside interface. Inbound transport of the packets is not allowed.	N/A
both	Allows inbound and outbound transport of packets of the specified protocol, application or service between inside and outside interfaces.	N/A

Example One - specifying a protocol <number>

The following example allows IGMP (Internet Group Management Protocol) packets inbound from the external interface to the DMZ interface. IGMP is protocol number 2 (see

http://www.ietf.org/rfc/rfc1700.txt).
First, we need to create a policy:
prompt> **firewall add policy ext-dmz external-dmz**
Then we can add the portfilter to it:
prompt> **firewall add portfilter pf1 ext-dmz protocol 2 inbound**

Example Two - specifying a TCP/UDP protocol
The following example allows DNS (Domain Name Service) outbound packets from the internal interface to the external interface. DNS uses UDP port 53 (see *http://www.ietf.org/rfc/rfc1700.txt*).
First, we need to create a policy:
prompt> **firewall add policy ext-int external-internal**
Then we can add the portfilter to it:
prompt> **firewall add portfilter pf2 ext-int udp 53 53 inbound**

Example Three - using a provided protocol, application or service
The following example allows SMTP (Simple Mail Transfer Protocol) packets inbound and outbound between the internal interface to the DMZ interface. This is a popular protocol that is provided by the Firewall. You do not need to specify the portnumber - the Firewall does this for you.
First, we need to create a policy:
prompt> **firewall add policy dmz-int dmz-internal**
Then we can add the portfilter to it:
prompt> **firewall add portfilter pf3 dmz-int smtp both**

1.106. firewall add validator

* This command is for future feature.

Syntax

firewall add validator <name> <policyname>
{inbound|outbound|both} <ipaddress> <hostipmask>

Description

Note - Before you can add validators to a Firewall policy, you must create a policy that determines how traffic is allowed/blocked, using the *allowonly-val* | *blockonly-val* options in the *firewall add policy* command:

- *allowonly-val*: only traffic based on the direction setting and the IP address(es) specified in the *firewall add validator* command is **allowed**. All other traffic is **blocked**.
- *blockonly-val*: only traffic based on the direction and the IP address(es) specified in the *firewall add validator* command is **blocked**. All other traffic is **allowed**. This command adds a validator to an existing Firewall policy. A validator allows/blocks traffic based on the source/destination IP address and netmask.

This command allows you to specify:

- the IP address(es) and netmask(s) that you want to allow/block
- the direction of traffic that you want to allow/block

Once you have added a validator to a policy, specifying the IP address and direction values, you can reuse these values by adding the validator to other policies.

Options

The following table gives the range of values for each option which can be specified with this command

and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the portfilter. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A
policyname	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A
inbound	Validator blocks incoming traffic based on IP addresses.	N/A
outbound	Validator blocks outgoing traffic based on IP addresses.	N/A
both	Validator filters inbound and outbound traffic based on IP addresses.	N/A
ipaddress	The IP address that you want to carry out IP address validation on. The IP address is displayed in the following format: 192.168.102.3	N/A
hostipmask	The IP mask address. If you want to filter a range of addresses, you can specify the mask, e.g., 255.255.255.0. If you want to filter a single IP address, you can use the specific IP mask address, e.g., 255.255.255.255.	N/A

Example

In the following example, a policy is created, then a validator added to block inbound and outbound traffic from/to the IP address stated. All other traffic is allowed.

prompt> **firewall add policy ext-int external-internal blockonly-val**
prompt> **firewall add validator v1 ext-int both 192.168.102.3 255.255.255.255**

1.107. firewall clear policies

Syntax

firewall clear policies

Description

This command deletes all existing policies from the firewall configuration. Any portfilters and validators associated with the policies are also deleted by this command.

Example

prompt> **firewall clear policies**

1.108. firewall clear portfilters

Syntax

firewall clear portfilters <policyname>

Description

This command deletes all portfilters that were added to an existing firewall policy using the *firewall add portfilter* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
policyname	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A

Example

prompt> **firewall clear portfilters ext-int**

1.109. firewall delete policy

Syntax

firewall delete policy <name>

Description

This command deletes a single existing policy from the firewall configuration. All portfilters and validators associated with the policy that you want to delete are also deleted by this command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A

Example

prompt> **firewall delete policy ext-dmz**

1.110. firewall delete portfilter

Syntax

firewall delete portfilter <name> <policyname>

Description

This command deletes a single portfilter that was added to a firewall policy using the *firewall add portfilter* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing portfilter. To display portfilter names, use the *firewall list portfilter* command.	N/A
policyname	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A

Example
prompt> **firewall delete portfilter pf3 ext-int**

1.111. firewall delete validator

* This command is for future feature.

Syntax
firewall delete validator <name> <policyname>
Description
This command deletes a single validator from a named policy.
Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing validator. To display validator names, use the *firewall list validators* command.	N/A
policyname	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A

Example
prompt> **firewall delete validator v1 ext-int**

1.112. firewall enable|disable

Syntax
firewall {enable | disable}
Description
This command enables/disables the entire Firewall module except for the IDS portion of the module.
Note - You **must** also enable the Security module, using the command *security*, if you want to use the Firewall module to configure security for your system. When the Firewall is enabled, all IP traffic on existing security interfaces that are NOT featured in a Firewall policy is blocked. For details on setting default policy security levels on security interfaces, see the *firewall set securitylevel* command. If you disable the Firewall during a session, any configuration changes made when the Firewall was enabled remain in the Firewall, so that you can re-enable them later in the session. If you need to reboot your system but want to save the Firewall configuration between sessions, use the *system config save* command.
Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enable	Enables the Firewall module.	disable
disable	Disables the Firewall module.	

Example
prompt> **firewall enable**

1.113. firewall enable|disable alerting email|paging

Syntax
Firewall {enable | disable} alerting {email | paging}
Description
This command enable/disable the email/pager firewall alerting information.
If enabled the email, open the outbound smtp port(25/tcp) in firewall policy.
If enabled the paging, open the outbound snpp port(444/tcp) in firewall policy.
Example
prompt> **firewall enable alerting email**

1.114. firewall enable|disable blockinglog

Syntax
firewall {enable | disable} blockinglog
Description
Note - To display logging information, you need to turn on *event logging* at the console. This command enables/disables whether Firewall blocking activity is logged.
Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enable	The blocking log is displayed.	enable
disable	The blocking log is not displayed.	

Example
prompt> **firewall enable blockinglog**

1.115. firewall enable|disable IDS

Syntax
firewall {enable | disable} IDS
Description
This command explicitly enables/disables the IDS (Intrusion Detection Service) portion of the Firewall. You must enable IDS if you want to activate the settings specified in the *firewall IDS* commands.
Note - You do not have to enable the Firewall module in order to use the IDS commands, however you **must** enable the Security module using the command *security*. If you disable IDS during a session, any configuration changes made when IDS was enabled remain in the Firewall, so that you can re-enable them later in the session.
Note - This CLI command is **case-sensitive**. You *must* type the command attributes exactly as they appear in the syntax section on this page. If you do not use the same case-sensitive syntax, the command fails and the CLI displays a syntax error message.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enable	Enables the IDS portion of the Firewall module.	disable
disable	Disables the IDS portion of the Firewall module.	

Example

prompt> **firewall enable IDS**

1.116. firewall enable|disable intrusionlog

Syntax

firewall {enable | disable} intrusionlog

Description

Note - To display logging information, you need to turn on *event logging* at the console. This command enables/disables whether details of attempted Firewall intrusion activity are logged.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enable	The intrusion log is displayed.	disable
disable	The intrusion log is not displayed.	

Example

prompt> **firewall enable intrusionlog**

1.117. firewall enable|disable sessionlog

Syntax

firewall {enable | disable} sessionlog

Description

Note - To display logging information, you need to turn on *event logging* at the console. This command enables/disables whether Firewall session events are logged.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enable	The log containing session details is displayed.	enable
disable	The log containing session details is not displayed.	

Example

prompt> **firewall enable sessionlog**

1.118. firewall list policies

Syntax

firewall list policies

Description

This command lists the following information about policies that were added to the firewall using the *firewall add policy* command:

- Policy ID number
- Policy name
- Interface Type 1 and Interface Type 2 - the two interface types between which a policy exists (external - internal, external - DMZ or internal - DMZ).
- Validator Allow Only status - *true* means that allowonly-val was set when the policy was created. *False* means that either blockonly-val was set, or no validator status was set (blockonly-val is the default setting if no status is specified).

Example

prompt> **firewall list policies**
Firewall Policies:
ID | Name | Type 1 | Type 2 | Validator Allow Only
---
1 | ext-dmz | external | dmz | true

1.119. firewall list portfilters

Syntax

firewall list portfilters <policyname>

Description

This command lists portfilters that were added to a firewall policy using the *firewall add portfilter* command. It displays the following information:

- Portfilter ID number
- Portfilter name
- Type - port number range or specified port number
- Port range used by the specified TCP or UDP protocol (e.g., 53 for DNS, 25 for SMTP). For non-TCP/UDP protocols, the port range is set to *0-0*.
- In - displays the inbound permission setting (true or false)
- Out- displays the outbound permission setting (true or false)
- Raw - displays whether or not the portfilter uses a non-TCP/UDP protocol (true or false)
- TCP - displays whether or not the portfilter uses a TCP protocol (true or false)
- UDP - displays whether or not the portfilter uses a UDP protocol (true or false)

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
policyname	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A

Example

prompt> **firewall list portfilters ext-int**

```
ID | Name | Type | Port Range | In | Out | Raw | TCP | UDP
---------------------------------------------------------------------------
1 | pf3 | 6 | 25 - 25 |true |true |false |true |false
2 | pf2 | 17 | 53 - 53 |false |true |false |false |true
3 | pf1 | 2 | 0 - 0 |true |false |true |false |false
---------------------------------------------------------------------------
```

1.120. firewall list protocol

Syntax

firewall list protocol

Description

This command will list some of the protocols associated with the tcp/ip suite. Useful for the non tcp/udp version of the port filter command.

Example

prompt> **firewall list protocol**

```
Assigned Internet Protocol Numbers
see RFC 1700 "Assigned Numbers"
section "Protocol Numbers" pages 7 - 9

  1   ICMP       Internet Control Message
  2   IGMP       Internet Group Management
  3   GGP        Gateway-to-Gateway
  4   IP         IP in IP (encapsulation)
  6   TCP        Transmission Control
  8   EGP        Exterior Gateway Protocol
  9   IGP        any private interior gateway
 17   UDP        User Datagram
 46   RSVP       Reservation Protocol
 47   GRE        General Routing Encapsulation
 89   OSPFIGP    OSPFIGP
 92   MTP        Multicast Transport Protocol
 94   IPIP       IP-within-IP Encapsulation Protocol
```

1.121. firewall list validators

* This command is for future feature.

Syntax

firewall list validators <policyname>

Description

This command lists the following information about validators added to a policy using the *firewall add validator* command:

- Validator ID number

- Validator name
- Direction (inbound, outbound or both)
- Host IP address
- Host mask address

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
policyname	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A

Example

prompt> **firewall list validators ext-int**

```
Firewall Host Validators:
ID | Name | Direction | Host IP          | Mask
-------------------------------------------------------------
2  | v1   | both      | 192.168.103.2 | 255.255.255.0
1  | v2   | inbound   | 192.168.103.1 | 255.255.255.0
-------------------------------------------------------------
```

1.122. firewall set alerting email server

Syntax

Firewall set alerting email server <email_server>

Description

This command allows you to set the alerting email (SMTP) server.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
email_server	IP address of email server	NULL

Example

prompt> **firewall set alerting email server 192.168.10.1**

1.123. firewall set alerting email from

Syntax

Firewall set alerting email from <from>

Description

This command allows you to specify the email address appearing on the from field.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value

from	Source address of the mail.	NULL

Example

prompt> **firewall set alerting email from 192.168.2.46**

1.124. firewall set alerting email recipient1

Syntax

Firewall set alerting email recipient1 <email><name>

Description

This command allows you to specify the email recipient 1: email address and recipient name.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
email	The email address of the email recipient1.	NULL
name	The email name of the email recipient1.	NULL

Example

prompt> firewall set alerting email recipient1 192.168.2.46 test

1.125. firewall set alerting email recipient2

Syntax

Firewall set alerting email recipient2 <email> <name>

Description

This command allows you to specify the email recipient 2: email address and recipient name.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
email	The email address of the email recipient2.	NULL
name	The email name of the email recipient2.	NULL

Example

prompt> firewall set alerting email recipient2 192.168.2.47 test1

1.126. firewall set alerting paging server

Syntax

Firewall set alerting paging server <paging_server>

Description

This command allows you to set the alerting paging (SNPP) server.

Options

The following table gives the range of values for each option which can be specified with this command

and a default value (if applicable).

Option	Description	Default value
paging_server	IP address of the paging server	NULL

Example

prompt> **firewall set alerting paging server 192.168.10.1**

1.127. firewall set alerting paging from

Syntax

Firewall set alerting paging from <from>

Description

This command allows you to specify the 'from' name.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
from	Source address of the paging.	NULL

Example

prompt> **firewall set alerting paging from 192.168.2.46**

1.128. firewall set alerting paging recipient1

Syntax

Firewall set alerting paging recipient1 <pager><name>

Description

This command allows you to specify the pager recipient 1: pager address and recipient name.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
pager	The address of the pager recipient1.	NULL
name	The name of the pager recipient1.	NULL

Example

prompt> firewall set alerting paging recipient1 192.168.2.46 test

1.129. firewall set alerting paging recipient2

Syntax

Firewall set alerting paging recipient2 <pager> <name>

Description

This command allows you to specify the pager recipient 2: pager address and recipient name.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
pager	The address of the pager recipient2.	NULL
name	The name of the pager recipient2.	NULL

Example

prompt> firewall set alerting paging recipient2 192.168.2.47 test1

1.130. firewall set IDS blacklist

Syntax

firewall set IDS blacklist {enable | disable | clear}

Description

This command sets the blacklist IDS (Intrusion Detection Setting). Blacklisting denies an external host access to the system if IDS has detected an intrusion from that host. Access to the network is denied for ten minutes.

Note - This CLI command is **case-sensitive**. You *must* type the command attributes exactly as they appear in the syntax section on this page. If you do not use the same case-sensitive syntax, the command fails and the CLI displays a syntax error message.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enable	Enables blacklisting of an external host if IDS has detected an intrusion from that host.	disable
disable	Disables blacklisting of an external host if IDS has detected an intrusion from that host.	
clear	Clears blacklisting of an external host.	

Example

prompt> **firewall set IDS blacklist enable**

1.131. firewall set IDS DOSattackblock

Syntax

firewall set IDS DOSattackblock <duration>

Description

This command sets the DOS (Denial of Service) attack block duration Intrusion Detection Setting (IDS). A DOS attack is an attempt by an attacker to prevent legitimate users from using a service. If a DOS attack is detected, all suspicious hosts are blocked by the firewall for a set time limit. This command allows you to specify the duration of the block time limit.

Note - This CLI command is **case-sensitive**. You *must* type the command attributes exactly as they appear in the syntax section on this page. If you do not use the same case-sensitive syntax, the command fails and the CLI displays a syntax error message.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
duration	The length of time (in seconds) that the firewall blocks suspicious hosts for once a DOS attack attempt has been detected by the firewall.	1800 (30 minutes)

Example

prompt> **firewall set DOSattackblock 3600**

1.132. firewall set IDS MaxICMP

Syntax

firewall set IDS MaxICMP <max>

Description

This command sets the maximum number of ICMP packets per second that are allowed by firewall before an ICMP Flood is detected. An ICMP Flood is a DOS (Denial of Service) attack. An attacker tries to flood the network with ICMP packets in order to prevent transportation of legitimate network traffic. Once the maximum number of ICMP packets per second is reached, an attempted ICMP Flood is detected. The firewall blocks the suspected attacker for the time limit specified in the *firewall set IDS DOSattackblock* command.

Note - This CLI command is **case-sensitive**. You *must* type the command attributes exactly as they appear in the syntax section on this page. If you do not use the same case-sensitive syntax, the command fails and the CLI displays a syntax error message.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
max	The maximum number (per second) of ICMP packets that are allowed before an ICMP Flood attempt is detected.	100

Example

prompt> **firewall set IDS MaxICMP 200**

1.133. firewall set IDS MaxPING

Syntax

firewall set IDS MaxPING <max>

Description

This command sets the maximum number of pings per second that are allowed by firewall before an Echo Storm is detected. Echo Storm is a DOS (Denial of Service) attack. An attacker sends oversized ICMP datagrams to the system using the 'ping' command. This can cause the system to crash, freeze or reboot, resulting in denial of service to legitimate users. Once the maximum number of pings per second is reached, an attempted DOS attack is detected. The firewall blocks the suspected attacker for the time limit specified in the *firewall set IDS DOSattackblock* command.

Note - This CLI command is **case-sensitive**. You *must* type the command attributes exactly as they appear in the syntax section on this page. If you do not use the same case-sensitive syntax, the command fails and the CLI displays a syntax error message.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
max	The maximum number (per second) of pings that are allowed before an Echo Storm attempt is detected.	15

Example

prompt> **firewall set IDS MaxPING 25**

1.134. firewall set IDS MaxTCPopenhandshake

Syntax

firewall set IDS MaxTCPopenhandshake <max>

Description

This command sets the maximum number of unfinished TCP handshaking sessions per second that are allowed by firewall before a SYN Flood is detected. SYN Flood is a DOS (Denial of Service) attack. When establishing normal TCP connections, three packets are exchanged:

1. A SYN (synchronize) packet is sent from the host to the network server
2. A SYN/ACK packet is sent from the network server to the host
3. An ACK (acknowledge) packet is sent from the host to the network server

If the host sends unreachable source addresses in the SYN packet, the server sends the SYN/ACK packets to the unreachable addresses and keeps resending them. This creates a backlog queue of unacknowledged SYN/ACK packets. Once the queue is full, the system will ignore all incoming SYN requests and no legitimate TCP connections can be established. Once the maximum number of unfinished TCP handshaking sessions is reached, an attempted DOS attack is detected. The firewall blocks the suspected attacker for the time limit specified in the *firewall set IDS DOSattackblock* command.

Note - This CLI command is **case-sensitive**. You *must* type the command attributes exactly as they appear in the syntax section on this page. If you do not use the same case-sensitive syntax, the command fails and the CLI displays a syntax error message.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
max	The maximum number (per second) of unfinished TCP handshaking sessions that are allowed before a SYN Flood attempt is detected.	100

Example

prompt> **firewall set IDS MaxTCPopenhandshake 150**

1.135. firewall set IDS SCANattackblock

Syntax

firewall set IDS SCANattackblock <duration>

Description

This command allows you to set the scan attack block duration Intrusion Detection Setting (IDS). The firewall detects when the system is being scanned by a suspicious host attempting to identify any open ports. If scan activity is detected, all suspicious hosts are blocked by the firewall for a set time limit. This command allows you to specify the duration of the block time limit.

Note - This CLI command is **case-sensitive**. You *must* type the command attributes exactly as they appear in the syntax section on this page. If you do not use the same case-sensitive syntax, the command fails and the CLI displays a syntax error message.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
duration	The length of time (in seconds) that the firewall blocks all suspicious hosts for, after it has detected scan activity on the Firewall.	86400 (one day)

Example

prompt> **firewall set IDS SCANattackblock 43200**

1.136. firewall set IDS victimprotection

Syntax

firewall set IDS victimprotection {enable <duration> | disable}

Description

This command enables/disables the victim protection Intrusion Detection Setting (IDS). Enabling this command protects the victim from an attempted spoofing attack. Web spoofing allows an attacker to create a 'shadow' copy of the World Wide Web. All access to the shadow Web goes through the attacker's machine, so the attacker can monitor all of the victim's activities and send false data to or from the victim's machine. If victim protection is enabled, packets destined for the victim host of a spoofing style attack are blocked. The command allows you to specify the duration of the block time limit.

Note - This CLI command is **case-sensitive**. You *must* type the command attributes exactly as they appear in the syntax section on this page. If you do not use the same case-sensitive syntax, the command fails and the CLI displays a syntax error message.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enable	Enables victim protection and blocks packets destined for the victim host.	disable
disable	Disables victim protection.	
duration	The length of time (in seconds) that the firewall blocks packets destined for the victim of a spoofing style attack.	600 (10 minutes)

Example

prompt> **firewall set IDS victimprotection enable 800**

1.137. firewall set privhost

Syntax

Firewall set privhost <privhost_start_addr> <privhost_end_addr>

Description

This command allows you to set the priviliege start and end hosts' IP address.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
privhost_start_addr	Privilege start IP address.	NULL
privhost_end_addr	Privilege end IP address.	NULL

Example

prompt> **firewall set privhost 200.199.241.1 200.199.241.254**

1.138. firewall set securitylevel

Syntax

firewall set securitylevel {none | high | medium | low | userdefined <slevel>}

Description

This command allows you to set which security level is used by the Firewall. There are three default security levels (high, medium and low) that contain different security configuration information for each interface connection. Once you have selected a security level, all IP traffic *except* the default policies specified will be blocked by the Firewall. The security level *none* blocks all IP traffic for every security interface. The *userdefined* option allows you to select a security configuration that you have previously created. There are three types of interface connections:

- Between the external interface and internal interface
- Between the external interface and the de-militarized zone (DMZ)
- Between the DMZ and the internal interface

Selecting a security level deletes the previous security level, and any policies or portfilters set, and replaces them with the newly selected level. You can add your own security policies using the *firewall add policy* command.

Options

The following tables describe the default policies enabled in the firewall for each of the high, medium and low security levels. The tables tell you whether a certain service can be received *in* or allowed *out* by a specific policy:

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable):

Option	Description	Default value
none	Your system blocks all IP traffic between interfaces.	none

high	Your system uses the *high* firewall security level, providing a high level of firewall security between interfaces.	
medium	Your system uses the *medium* firewall security level, providing a medium level of firewall security between interfaces.	
low	Your system uses the *low* firewall security level, providing a low level of firewall security between interfaces.	
userdefined	Your system uses a security configuration that you have previously created.	
slevel	The name of the security configuration level that you have previously created	N/A

Example

prompt> **firewall set securitylevel medium**

1.139. firewall show alerting

Syntax

Firewall show alerting

Description

This command displays the firewall alerting information.

Example

prompt> **firewall show alerting**

```
Alerting:

    Email (SMTP)
        enabled:                    false
        server:
        from:
        recipient 1
            name:
            email:
        recipient 2
            name:
            email:

    Paging (SNPP)
        enabled:                    false
        server:
        from:
        recipient 1:
        recipient 2:
```

1.140. firewall show IDS

Syntax

firewall show IDS

Description

This command displays the following information about the Firewall IDS settings:

- IDS enabled status (true or false)
- Blacklist status (true or false)
- Use Victim Protection status (true or false)
- DOS attack block duration (in seconds)
- Scan attack block duration (in seconds)
- Victim protection block duration (in seconds)
- Maximum TCP open handshaking count allowed (per second)
- Maximum ping count allowed (per second)
- Maximum ICMP count allowed (per second)

Example

prompt> **firewall show IDS**
Firewall IDS:
IDS Enabled: true
Use Blacklist: true
Use Victim Protection: true
Dos Attack Block Duration: 1800
Scan Attack Block Duration: 10
Victim Protection Block Duration: 600
Max TCP Open Handshaking Count: 100
Max PING Count: 20
Max ICMP Count: 100

1.141. firewall show policy

Syntax

firewall show policy <name>

Description

This command displays information about a single policy that was added to the firewall using the *firewall add policy* command. A policy exists between two interface types that were set using the *firewall add policy* command. This command displays what these interface types are, and the allow only validator status; *true* means that allowonly-val was set when the policy was created; *false* means that either blockonly-val was set, or no validator status was set (blockonlyval is the default setting if no status is specified).

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A

Example

prompt> **firewall show policy p2**
Firewall Policy: ext-dmz
Interface Type 1: external
Interface Type 2: dmz
Allow Only Validator: true

1.142. firewall show portfilter

Syntax

firewall show portfilter <name> <policyname>

Description

This command displays information about a single portfilter that was added to a firewall policy using the *firewall policy add portfilter* command. The following portfilter information is displayed:

- Portfilter name
- Transport type used by the protocol (e.g., 6 for SMTP)
- Start of the port range
- End of the port range
- Inbound permission (true or false)
- Outbound permission (true or false)
- Raw IP - whether the portfilter uses a non-TCP/UDP protocol (true or false)
- TCP permission - whether the portfilter uses a TCP protocol (true or false)
- UDP permission - whether the portfilter uses a UDP protocol (true or false)

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing portfilter. To display portfilter names, use the *firewall list portfilters* command.	N/A
policyname	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A

Example

prompt> **firewall show portfilter pf3 ext-int**
Firewall Port Filter: pf3
Transport type: 6
Port number start: 25
Port number end: 25
Inbound permission: true
Outbound permission: true
Raw IP: false
TCP permission: true
UDP permission: false

1.143. firewall show privhost

Syntax

Firewall show privhost

Description

This command displays the priviliege start and end hosts' IP address.

Example

prompt> **firewall show privhost**
Priviledge Host Start: 200.199.241.1
Priviledge Host End: 200.199.241.254

1.144. firewall show validator

* This command is not useful at present.

Syntax

firewall show validator <name> <policyname>

Description

This command displays information about a single validator that was added to firewall policy using the *firewall add validator* command. The following validator information is displayed:

- Validator name
- Direction (inbound, outbound or both)
- Host IP address
- Host mask address

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing validator. To display validator names, use the *firewall list validators* command.	N/A
policyname	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A

Example

prompt> **firewall show validator v1**
Firewall Host Validator: v1
Direction: both
Host IP: 192.168.103.2
Host Mask: 255.255.255.0

1.145. firewall status

Syntax

firewall status

Description

This command displays the following information about the Firewall:

- Firewall status (enabled or disabled)
- Security level setting (none, high, low or medium)
- Firewall logging status:
- session logging (enabled or disabled)
- blocking logging (enabled or disabled)
- intrusion logging (enabled or disabled)

Example

prompt> **firewall status**
Firewall enabled.
Firewall security level: medium.
Firewall session logging enabled.
Firewall blocking logging enabled.
Firewall intrusion logging disabled.

Frame Relay CLI commands
This chapter describes the Frame Relay CLI commands.

1.146. framerelay add transport

Syntax
framerelay add transport <name> <port> <dlci>

Description
This command adds a named Frame Relay transport and allows you to specify which port it will use to transport Frame Relay data and specify a Data Link Channel Identifier (DLCI) to identify the Frame Relay channel that you are using.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the Frame Relay transport. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A
port	The port is used to transport frame relay data.	N/A
dlci	A number that specifies the PVC or SVC in a Frame Relay network. The DLCI can be any positive number less than 8196.	N/A

Example
prompt> **framerelay add transport fr1 fr_relay 171**

1.147. framerelay clear transports

Syntax
framerelay clear transports

Description
This command deletes all Frame Relay transports that were created using the *framerelay add transport* command.

Example
prompt> **framrelay clear transports**

1.148. framerelay delete transport

Syntax
framerelay delete transport {<name>|<number>}

Description
This command deletes a single Frame Relay transport.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing Frame Relay transport. To display transport names, use the *framerelay list transports* command.	N/A

number	A number that identifies an existing Frame Relay transport. To display transport numbers, use the *framerelay list transports* command.	N/A

Example

prompt> **framerelay delete transport fr1**

1.149. framerelay list transports

Syntax

framerelay list transports

Description

This command lists all Frame Relay transports that have been created using the *framerelay add transport* command. It displays the transport identification number and name, and the name of the port that it uses to transport Frame Relay data.

Example

prompt> **framerelay list transports**

```
Frame Relay Transports:
ID | Name | Port        | DLCI | Encapsulation
----|------ --|------------|--------|--------------------
1    | fr1      | fr_relay | 171    | Raw
-------------------------------------------------------
```

1.150. framerelay set transport chnlsegmentsize

Syntax

framerelay set transport {<name>|<number>} chnlsegmentsize
<channel segment size>

Description

This command sets the size of the channel segment used by Frame Relay.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing Frame Relay transport. To display transport names, use the *framerelay list transports* command.	N/A
number	A number that identifies an existing Frame Relay transport. To display transport numbers, use the *framerelay list transports* command.	N/A
Chnlsegment size	The size of the channel segment used by Frame Relay. If you set this to any number other than 0, DLCI level FRF.12 segmentation is enabled.	0

Example

prompt> **framerelay set transport fr1 chnlsegmentsize 50**

1.151. framerelay set transport dlci

Syntax

framerelay set transport {<name>|<number>} dlci <dlci>

Description

This command sets the DLCI; the identifier for the Frame Relay data link channel that you are using.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing Frame Relay transport. To display transport names, use the *framerelay list transports* command.	N/A
number	A number that identifies an existing Frame Relay transport. To display transport numbers, use the *framerelay list transports* command.	N/A
dlci	A number that specifies the PVC or SVC in a Frame Relay network. The DLCI can be any positive number less than 8196.	N/A

Example

prompt> **framerelay set transport fr1 dlci 80**

1.152. framerelay set transport encapsulation

Syntax

framerelay set transport {<name>|<number>} encapsulation {raw|routedip|bridgedether}

Description

This command sets the RFC1490 encapsulation method used by Frame Relay. Each DLCI can be multiplexed further if you are using RFC1490 multiprotocol encapsulation.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing Frame Relay transport. To display transport names, use the *framerelay list transports* command.	N/A
number	A number that identifies an existing Frame Relay transport. To display transport numbers, use the *framerelay list transports* command.	N/A
raw	No RFC1490 encapsulation.	raw
routedip	RFC1490 encapsulation is used and IP packets are routed over Frame Relay.	
bridgedether	RFC1490 encapsulation is used and Ethernet packets are bridged over Frame Relay.	

Example

prompt> **framerelay set transport encapsulation bridgedether**

1.153. framerelay set transport port

Syntax

framerelay set transport {<name>|<number>} port <port>

Description

This command sets the port that an existing Frame Relay transport uses to transport data.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing Frame Relay transport. To display transport names, use the *framerelay list transports* command.	N/A
number	A number that identifies an existing Frame Relay transport. To display transport numbers, use the *framerelay list transports* command.	N/A
port	The port is used to transport Frame Relay data.	N/A

Example

prompt> **framerelay set transport fr1 port fr_relay**

1.154. framerelay set transport rxmaxpdu

Syntax

framerelay set transport {<name>|<number>} rxmaxpdu <rxmaxpdu>

Description

This command sets the maximum Protocol Data Unit (PDU) size that can be received over Frame Relay.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing Frame Relay transport. To display transport names, use the *framerelay list transports* command.	N/A
number	A number that identifies an existing Frame Relay transport. To display transport numbers, use the *framerelay list transports* command.	N/A
rxmaxpdu	The maximum size of protocol data units that Frame Relay can receive.	8192

Example

prompt> **framerelay set transport fr1 rxmaxpdu 10000**

1.155. framerelay set transport tcmaxpdu

Syntax

framerelay set transport {<name>|<number>} tcmaxpdu <tcmaxpdu>

Description

This command sets the maximum Protocol Data Unit (PDU) size that can be transmitted over Frame Relay.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value

name	A name that identifies an existing Frame Relay transport. To display transport names, use the *framerelay list transports* command.	N/A
number	A number that identifies an existing Frame Relay transport. To display transport numbers, use the *framerelay list transports* command.	N/A
tcmaxpdu	The maximum size of protocol data units that Frame Relay can transmit.	8192

Example

prompt> **framerelay set transport fr1 txmaxpdu 10000**

1.156. framerelay show transport

Syntax

framerelay show transport {<name>|<number>}

Description

This command displays the following information about a single Frame Relay transport:

- Transport name
- Transport description
- Frame Relay port
- DLCI setting
- Encapsulation method
- RX Max PDU setting
- TX Max PDU setting
- Segment size

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing Frame Relay transport. To display transport names, use the *framerelay list transports* command.	N/A
number	A number that identifies an existing Frame Relay transport. To display transport numbers, use the *framerelay list transports* command.	N/A

Example

prompt> **framerelay show transport fr1**
Frame Relay Transport: fr1
Description: fr1
Port: fr_relay
DLCI: 171
Encapsulation: RoutedIP
RX Max PDU: 10000
TX Max PDU: 10000
Segment size: 50

IGMP CLI commands
This chapter describes the Internet Group Management Protocol (IGMP) CLI commands.

Note – The router **must** be the IGMP Querier in order to forward packets; this may restrict the IP addressing scheme used. In particular, the IGMP-based forwarding router must be given the lowest IP addresses of any potential IGMP Queriers on the link, in order to win the IGMP Querier election. If another device wins the IGMP Querier election, no packets will follow.

1.157. igmp set upstreaminterface

Syntax

igmp set upstreaminterface {<ip_interface> | none}

Description

This command enables the router's IGMP Proxy, and sets one of the router's existing IP interfaces as the upstream interface; all other router interfaces are designated downstream interfaces. The upstream interface implements the *Host* portion of the IGMP protocol, and the downstream interfaces implement the *Router* portion of the IGMP protocol. The IGMP Proxy may be disabled by setting upstream interface to *none*.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
ip_interface	The name of an existing router interface that you want to set as the upstreaminterface.	N/A
none	Disables IGMP proxy	N/A

Example

prompt> **igmp set upstreaminterface ip1**

1.158. igmp show upstreaminterface

Syntax

igmp show upstreaminterface

Description

This command displays the status of the upstream interface. If an upstream interface has been set using the *igmp set upstreaminterface* command, this command displays the current setting.

Example

prompt> **igmp show upstreaminterface**
IGMP Proxy configuration
Upstream If : ip1

1.159. igmp show status

Syntax

igmp show status

Description

This command displays the following information about the status of IGMP:

- IGMP Proxy group membership per interface details
- Interface name and querier status
- Group address

Example

prompt> **igmp show status**

```
Multicast group membership:
Interface (querier) | Group address
--------------------|----------------
eth0 (yes)          | 239.255.255.250
---------------------------------------
```

IPSEC CLI commands

1.160. ipsec add endpoint

Syntax

ipsec add endpoint <endpoint_id>

Description

This command allows you to add an ipsec endpoint.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
endpoint_id	ID of the endpoint (termination VPN router). IKE defines two modes when negotiating a phase 1 SA: main mode and aggressive mode. For Aggressive Mode use a string like main@ABCD.com. For Main Mode use the WAN IP address of the enpoint VPN router.	N/A

Example

ipsec add endpoint main@ABCD.com

1.161. ipsec clear endpoints

Syntax

ipsec clear endpoint

Description

This command allows you to delete all ipsec endpoints.

Example

ipsec clear endpoint

1.162. ipsec delete endpoint

Syntax

ipsec delete endpoint <endpoint_id>

Description

This command allows you to delete an ipsec endpoint.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
endpoint_id	ID of the endpoint.	N/A

Example

ipsec delete endpoint 1

1.163. ipsec list endpoints

Syntax

ipsec list endpoints

Description

This command allows you to list all ipsec endpoints.

Example

ipsec list endpoints

```
IPSec Endpoints:
-----------------------------------------------------------------------------------------------
  No  |            ID              |  IP address      |   Status
      |        Target Host         |   Range?         |   Bytes Sent/Received
-----------------------------------------------------------------------------------------------
   1  |             1              |   0.0.0.0        |   broken
      |       0.0.0.0/0.0.0.0      |   true           |     0 / 0
-----------------------------------------------------------------------------------------------
```

1.164. ipsec set endpoint endpointid

Syntax

ipsec set endpoint <number> endpointid <endpoint_id>

Description

This command allows you to set the number and endpoint id of the ipsec.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A
endpoint_id	Endpoint id.	N/A

Example

ipsec set endpoint 5 endpointid 1

1.165. ipsec set endpoint ike auth digital-signature

Syntax

ipsec set endpoint <number> ike auth digital-signature

Description

This command allows you to set the ike authentication method to ‘digital-signature‘ of the ipsec endpoint.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A

Example

ipsec set endpoint 5 ike auth digital-signature

1.166. ipsec set endpoint ike auth preshared-key

Syntax

ipsec set endpoint <number> ike auth preshared-key

Description

This command allows you to set the ike authentication method to ‘preshared-key ‘of the ipsec endpoint.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A

Example

ipsec set endpoint 5 ike auth preshared-key

1.167. ipsec set endpoint ike encryption

Syntax

ipsec set endpoint <number> ike encryption{des | blowfish | 3des}

Description

This command allows you to set encryption algorithm of the ipsec endpoint. The choices are: des, blowfish, 3des. Encryption is a mathematical operation that transforms data from “clear text” (something that a human or a program can interpret) to “cipher test” (something that cannot be interpreted). Usually the mathematical operation requires that an alphanumeric “key” be supplied along with the clear text. The key and clear text are processed by the encryption operation which leads to the data scrambling that makes encryption secure.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A

Example

ipsec set endpoint 5 ike encryption des

1.168. ipsec set endpoint ike hash

Syntax

ipsec set endpoint <number> ike hash {md5|sha1}

Description

This command allows you to set the IKE hash algorithm of the ipsec endpoint.. The choices are: md5, sha1. When a certificate is issued by a provider, it is not generally the overall certificate but a cryptographic check sum from the certificate that is signed. The procedure used for calculating

the check sum is referred to as a hash algorithm.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A

Example

ipsec set endpoint 5 ike hash md5

1.169. ipsec set endpoint ike presharedkey

Syntax

ipsec set endpoint <number> ike presharedkey <preshared_key>

Description

This command allows you to set the IKE pre-shared key. The key takes effective when pre-shared key is set as the authentication method.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A
preshared_key	IKE pre-shared key	N/A

Example

ipsec set endpoint 5 ike presharedkey ABCD

1.170. ipsec set endpoint ipaddress

Syntax

ipsec set endpoint <number> ipaddress <ip_address>

Description

This command allows you to set the termination ip address of the ipsec endpoint.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A
ip_address	IP Address.	N/A

Example

ipsec set endpoint 5 ipaddress 192.168.10.1

1.171. ipsec set endpoint ipsec ah

Syntax

ipsec set endpoint <number> ipsec ah <ah_transform>

Description

This command allows you to set the IPSec AH transform of the ipsec endpoint. The choices are: md5, sha1, des-mac. The Authentication Header is a mechanism for providing strong integrity and authentication for IP packets.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A
ah_transform	AH transform. The choices are: md5, sha1, des-mac.	N/A

Example

ipsec set endpoint 5 ipsec ah md5

1.172. ipsec set endpoint ipsec esp

Syntax

ipsec set endpoint <number> ipsec esp <esp_transform>

Description

This command allows you to set the IPSec ESP transform of the ipsec endpoint. The choices are: des, 3des, blowfish, rc4, esp-null.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A
esp_transform	ESP transform. The choices are: des, 3des, blowfish, rc4, esp-null.	N/A

Example

ipsec set endpoint 5 ipsec esp des

1.173. ipsec set endpoint ipsec esp_auth

Syntax

ipsec set endpoint <number> ipsec esp_auth <esp_auth>

Description

This command allows you to set the IPSec ESP auth algorithm of the ipsec endpoint. The choices are: md5, sha1, des-mac, null.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value

number	Number of the endpoint.	N/A
esp_auth	ESP auth algorithm. The choices are: md5, sha1, des-mac, null.	N/A

Example

ipsec set endpoint 5 ipsec esp_auth md5

1.174. ipsec set endpoint ipsec ipcomp

Syntax

ipsec set endpoint <number> ipsec ipcomp <ipcomp_auth>

Description

This command allows you to set the IPSec IPCOMP transform of the ipsec endpoint. The choices are: lzs. IPCOMP(IP payload compression) is a protocol to reduce the size of IP datagrams.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A
ipcomp_auth	IPCOMP transform. The choices are: lzs.	N/A

Example

ipsec set endpoint 5 ipsec ipcomp_auth **lzs**

1.175. ipsec set endpoint ipsec protocol

Syntax

ipsec set endpoint <number> ipsec protocol <protocol_type>

Description

This command allows you to set the IPSec protocol of the ipsec endpoint. The choices are: esp, ah, ipcomp, ah-esp, ah-ipcomp, esp-ipcomp.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A
protocol_type	Protocol type. The choices are: esp, ah, ipcomp, ah-esp, ah-ipcomp, esp-ipcomp.	N/A

Example

ipsec set endpoint 5 ipsec protocol esp

1.176. ipsec set endpoint ipsec tunnel_type

Syntax

ipsec set endpoint <number> ipsec tunnel_type <tunnel_type>

Description

This command allows you to set the IPSec tunnel type of the ipsec endpoint. The choices are: public,

private.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A
tunnel_type	Tunnel type. The choices are: public, private. Public uses the ESP protocol only. Private provides UDP encapsulation for NAT traversal. We are using ports 2787 (ESP), 2788 (AH), and 2845 (IPCOMP). Public should be used for initial testing.	N/A

Example

ipsec set endpoint 5 ipsec tunnel_type private

1.177. ipsec set endpoint salife

Syntax

ipsec set endpoint <number> salife <seconds>

Description

This command allows you to set the SA life time which is the actual length of the security association's (SA) life, in seconds, of the ipsec endpoint.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A
seconds	The SA life time, in seconds	N/A

Example

ipsec set endpoint 5 salife 10

1.178. ipsec set endpoint target_host range

Syntax

ipsec set endpoint <number> target_host range <ip_address_start> <ip_address_end>

Description

This command allows you to set the IP range based target host for the IPsec. Please specify the start and end ip address.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A
ip_address_start	Start IP address.	N/A
ip_address_end	End IP address.	N/A

Example

ipsec set endpoint 5 target_host range 192.168.10.1 192.168.10.254

1.179. ipsec set endpoint target_host subnet

Syntax

ipsec set endpoint <number> target_host subnet < ip_address> <subnet_mask>

Description

This command allows you to set the subnet based target host for the IPsec. Please specify the ip address, subnet mask.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A
ip_address	IP address.	N/A
Subnet_mask	Subnet mask.	N/A

Example

ipsec set endpoint 5 target_host subnet 192.168.10.1 255.255.255.0

1.180. ipsec set intranet

Syntax

ipsec set intranet <intranet_addr><intranet_mask>

Description

This command allows you to set the intranet for ipsec.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
intranet_addr	IP Address of the intranet.	N/A
intranet_mask	Mask of the intranet.	N/A

Example

ipsec set intranet 172.168.2.128 255.255.255.255

1.181. ipsec set negotiationid

Syntax

ipsec set negotiationid <negotiation_id>

Description

This command allows you to set the negotiaiton id for ipsec.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
negotiation_id	The negotiation id for ipsec. For Aggressive Mode use a string like remote@ABCD.com. For Main Mode use the WAN IP address of the VPN router.	N/A

Example

Ipsec set negotiationid remote@ABCD.com

1.182. ipsec show endpoint

Syntax

ipsec show endpoint <number>

Description

This command allows you to display the detail information about the ipsec endpoint.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
number	Number of the endpoint.	N/A

Example

ipsec show endpoint 1

1.183. ipsec show intranet

Syntax

ipsec show intranet

Description

This command allows you to show the intranet for ipsec.

Example

ipsec show intranet
Intranet Ip Addr: 192.168.0.0
Intranet Ip Mask: 255.255.255.0

1.184. ipsec show negotiationid

Syntax

ipsec show negotiationid

Description

This command allows you to show the negotiation id for ipsec.

Example

ipsec show negotiationid
Negotiation Id: 200.200.200.1

L2TP CLI commands

1.185. anscl2tp set pool

Syntax

anscl2tp set pool <pool_start_addr> <pool_end_addr>

Description

This command allows you to set the starting and ending IP address of the l2tp ip pool.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
pool_start_addr	Starting IP Address of the pptp ip pool.	N/A
pool_end_addr	Ending IP Address of the pptp ip pool.	N/A

Example

anscl2tp set pool 172.168.3.128 172.168.3.191

1.186. anscl2tp show pool

Syntax

Anscl2tp show pool

Description

This command shows the IP pool of ansc l2tp.

Example

--> anscl2tp show pool
Ip Pool Start: 172.168.3.128
Ip Pool End: 172.168.3.191

1.187. anscl2tp show client

Syntax

anscl2tp show client

Description

This command shows connected clients.

Example

--> anscl2tp show client
Client List:

NAT CLI commands
This chapter describes the NAT (Network Address Translation) CLI commands.

1.188. nat add globalpool

Syntax
nat add globalpool <name> <interfacename> {internal|dmz}
<ipaddress> {subnetmask <mask>|endaddress <address>}

Description
The *nat enable* command creates an IP address for the outside security interface, however, you may want to use more than one outside IP address. For example, if your ISP provides multiple IP addresses, you might want to map an outside address to an inside interface that is your web server, and map another outside address to an inside interface that is your mail server. This command creates a pool of outside network addresses. A network address pool is a range of IP addresses that is visible outside your network. NAT translates packets between the outside addresses and the inside interfaces that each address is mapped to. There are two ways to specify a range of IP addresses:
1 specify the interfacename IP address and a subnet mask address
2 specify the interfacename IP address that represents the first address in the range, then specify the last address in the range
If you want to map IP addresses to individual hosts on an inside interface type, you can use the command *nat add resvmap globalip.*

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies a global network address or pool of addresses. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A
interfacename	The name of an existing security interface (external or DMZ) created and connected to an inside interface (DMZ or internal) using the *nat enable* command. To display security interfaces, use the *security list interfaces* command.	N/A
internal	Maps the IP addresses to the internal interface type inside the network.	N/A
dmz	Maps the global addresses to the DMZ interface type inside the network.	N/A
ipaddress	The IP address of the *interfacename* that is visible outside the network.	N/A
mask	The subnet mask of the network IP address.	N/A
endaddress	The last IP address in the range of addresses that make up the global address pool.	N/A

Examples
Example 1
This example creates a network address pool that allows NAT to translate packets between the external

interface and the DMZ interface type.
First, NAT is enabled between the external interface and the DMZ interface type:
prompt> **nat enable n1 extinterface dmz**
Then the IP address and subnet mask is created:
prompt> **nat add globalpool gp1 extinterface dmz**
192.168.102.3 subnetmask 255.255.255.0

Example 2
This example creates a network address pool that allows NAT to translate packets between the external interface and the internal interface type. First NAT is enabled between the external interface and the internal interface type:
prompt> **nat enable n2 extinterface internal**
Then the address range is created:
prompt> **nat add globalpool gp2 extinterface internal**
192.168.103.2 endaddress 192.168.103.50

1.189. nat add resvmap globalip

Syntax

nat add resvmap <name> globalip <interfacename> <globalip> <internalip> {tcp <portno>|udp <portno>|icmp|igmp|ip|egp|rsvp|ospf|ipip|all}

Description

This command maps an IP address from a global pool (created using the *nat add globalpool* command) to an individual IP address inside the network. NAT translates packets between the outside IP address and the individual host based on the transport information given in this command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing global IP address. To display global IP addresses, use the *nat list globalpools* command.	N/A
interfacename	The name of an existing security interface (external or DMZ) created and connected to an inside interface (DMZ or internal) using the *nat enable* command. To display security interfaces, use the *security list interfaces* command.	N/A
globalip	The IP address of an outside interface set using the *nat add globalpool* command.	N/A
internalip	The IP address of an individual host inside the network (internal or DMZ interface type).	N/A
(tcp) portno	The TCP port number that you want to use in your reserved mapping configuration.	N/A
(udp) portno	The UDP port number that you want to use in your reserved mapping configuration.	N/A

icmp	Internet Control Message Protocol (ICMP) is set as the transport type. ICMP messages are used for out-of-band messages related to network operation or mis-operation. See *http://www.ietf.org/rfc/rfc0792.txt*.	N/A
igmp	Internet Group Management Protocol (IGMP) is set as the transport type. Allows Internet hosts to participate in multicasting. See *http://www.ietf.org/rfc/rfc1112.txt*.	N/A
ip	Internetwork Protocol (IP). Provides all of the Internet's data transport services. *http://www.ietf.org/rfc/rfc791.txt* and *http://www.ietf.org/rfc/rfc919.txt*.	N/A
egp	Exterior Gateway Protocol (EGP).Protocol for exchanging routing information between autonomoussystems. See *http://www.ietf.org/rfc/rfc904.txt*.	N/A
rsvp	Resource Reservation Protocol (RSVP) is set as the transport type. Supports the reservation of resources across an IP network. See *http://www.ietf.org/rfc/rfc2205.txt*.	N/A
ospf	Open Shortest Path First (OSPF) is set as the transport type. A link-state routing protocol. See *http://www.ietf.org/rfc/rfc1583*.	N/A
ipip	IP-within-IP Encapsulation Protocol. Encapsulates an IP datagram within a datagram. See *http://www.ietf.org/rfc/rfc2896.txt*.	N/A
all	All traffic is translated between the global IP address and the specified inside address that it is mapped to.	N/A

Example

prompt> **nat add resvmap rm1 globalip extinterface 192.168.68.68 10.10.10.10 tcp 25**

1.190. nat add resvmap interfacename

Syntax

nat add resvmap <name> interfacename <interfacename> <internalip> {tcp <portno>|udp <portno>|icmp|igmp|ip|egp|rsvp|ospf|ipip|all}

Description

This command maps an outside IP security interface (enabled as a NAT object using the *nat enable* command) to an individual IP address inside the network. NAT translates packets between the outside IP address and the individual host based on the transport information given in this command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value

name	A name that identifies an existing global IP address. To display global IP addresses, use the *nat list globalpools* command.	N/A
interfacename	The name of an existing security interface (external or DMZ) created and connected to an inside interface (DMZ or internal) using the *nat enable* command. To display security interfaces, use the *security list interfaces* command.	N/A
globalip	The IP address of an outside interface set using the *nat add globalpool* command.	N/A
internalip	The IP address of an individual host inside the network (internal or DMZ interface type).	N/A
(tcp) portno	The TCP port number that you want to use in your reserved mapping configuration.	N/A
(udp) portno	The UDP port number that you want to use in your reserved mapping configuration.	N/A
icmp	Internet Control Message Protocol (ICMP) is set as the transport type. ICMP messages are used for out-of-band messages related to network operation or mis-operation. See *http://www.ietf.org/rfc/rfc0792.txt*.	N/A
igmp	Internet Group Management Protocol (IGMP) is set as the transport type. Allows Internet hosts to participate in multicasting. See *http://www.ietf.org/rfc/rfc1112.txt*.	N/A
ip	Internetwork Protocol (IP). Provides all of the Internet's data transport services. *http://www.ietf.org/rfc/rfc791.txt* and *http://www.ietf.org/rfc/rfc919.txt*.	N/A
egp	Exterior Gateway Protocol (EGP).Protocol for exchanging routing information between autonomoussystems. See *http://www.ietf.org/rfc/rfc904.txt*.	N/A
rsvp	Resource Reservation Protocol (RSVP) is set as the transport type. Supports the reservation of resources across an IP network. See *http://www.ietf.org/rfc/rfc2205.txt*.	N/A
ospf	Open Shortest Path First (OSPF) is set as the transport type. A link-state routing protocol. See *http://www.ietf.org/rfc/rfc1583*.	N/A
ipip	IP-within-IP Encapsulation Protocol. Encapsulates an IP datagram within a datagram. See *http://www.ietf.org/rfc/rfc2896.txt*.	N/A
all	All traffic is translated between the global IP address and the specified inside address that it is mapped to.	N/A

Example

prompt> nat add resvmap rm1 interfacename extinterface 10.10.10.10 tcp 25

1.191. nat clear globalpools

Syntax

nat clear globalpools <interfacename>

Description

This command deletes all address pools that were added to a specific outside interface using the *nat add globalpool* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
interfacename	The name of an existing security interface (external or DMZ) created and connected to an inside interface (DMZ or internal) using the *nat enable* command. To display security interfaces, use the *security list interfaces* command.	N/A

Example

prompt> **nat clear globalpools extinterface**

1.192. nat clear resvmaps

Syntax

nat clear resvmaps <interfacename>

Description

This command deletes all NAT reserved mappings that were added to an outside security interface using the *nat add resvmap* commands.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
interfacename	The name of an existing security interface (external or DMZ) created and connected to an inside interface (DMZ or internal) using the *nat enable* command. To display security interfaces, use the *security list interfaces* command.	N/A

Example

prompt> **nat clear resvmaps extinterface**

1.193. nat delete globalpool

Syntax

nat delete globalpool <name> <interfacename>

Description

This command deletes a single address pool that was added to a specific outside interface using the *nat add globalpool* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing global IP address. To display global IP addresses, use the *nat list globalpools* command.	N/A
interfacename	The name of an existing security interface (external or DMZ) created and connected to an inside interface (DMZ or internal) using the *nat enable* command. To display security interfaces, use the *security list interfaces* command.	N/A

Example

prompt> **nat delete globalpool gp1 extinterface**

1.194. nat delete resvmap

Syntax

nat delete resvmap <name> <interfacename>

Description

This command deletes a single NAT reserved mapping that was added to an outside security interface using the *nat add resvmap* commands.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing global IP address. To display global IP addresses, use the *nat list globalpools* command.	N/A
interfacename	The name of an existing security interface (external or DMZ) created and connected to an inside interface (DMZ or internal) using the *nat enable* command. To display security interfaces, use the *security list interfaces* command.	N/A

Example

prompt> **nat delete resvmap rm1 extinterface**

1.195. nat disable

Syntax

nat disable <name>

Description

This command disables a NAT object that was previously enabled between an existing security interface and a network interface type using the *nat enable* command. NAT is disabled between the security interface and all the interfaces that belong to the chosen interface type.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	The name of an existing NAT object created between a security interface and an interface type using the *nat enable* command. To display enabled NAT objects, use the *nat status* command.	N/A

Example

prompt> **nat disable nat1**

1.196. nat enable

Syntax

nat enable <name> <interfacename> {internal|dmz}

Description

This command enables NAT between an existing security interface and a network interface type. NAT is enabled between the security interface and all the interfaces that belong to the chosen network interface type. An interface is either an *inside* or *outside* interface. The network attached to an inside interface needs to be protected from the network attached to an outside interface. For example, the network attached to an internal interface (inside) needs to be protected from the network attached to a DMZ (outside). Also, you can only enable NAT between two different interface types. For example, if *interfacename* is an external interface type, you can enable NAT between the *interfacename* and the internal or the DMZ interface type, but not the external interface type. The following interface combinations are the only ones that you can use:

- external (outside) and internal (inside)
- external (outside) and DMZ (inside)
- DMZ (outside) and internal (inside)

The existing security interface must be an outside interface. NAT translates packets between the outside interface and the inside interface type. In this way, the IP address of a host on a network attached to an inside interface is hidden from a host on a network attached to an outside interface. If you want to map an outside interface to an individual host on an inside interface type, you can use the command *nat add resvmap interfacename*.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies a NAT object enabled between a security interface and an interface type. It can be made up of one or more	N/A

	letters or a combination of letters and digits, but it cannot start with a digit.	
interfacename	The name of an existing security interface (external or DMZ) that was added to the Security package using the *security add interface* command. To display security interfaces, use the *security list interfaces* command.	N/A
internal	Allows NAT to be enabled/disabled between the *interfacename* and all interfaces that belong to the *internal* interface type.	N/A
dmz	Allows NAT to be enabled/disabled between the *interfacename* and all interfaces that belong to the *DMZ* interface type. The *interfacename* must be an external interface type.	N/A

Example

prompt> **nat enable nat1 extinterface internal**

1.197. nat list globalpools

Syntax

nat list globalpools <interfacename>

Description

This command lists the following NAT address pool information for a specific outside interface:

• Address pool identification number
• Address pool name
• Type of inside interface (internal or DMZ)
• Subnet status (true or false)
• IP address - the outside network IP address or the first address in the range of network pool addresses
• Mask/End Address - the outside subnet mask of the outside network IP address or the last address in the range of network pool addresses

19.8.3 Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
interfacename	The name of an existing security interface (external or DMZ) created and connected to an inside interface (DMZ or internal) using the *nat enable* command. To display security interfaces, use the *security list interfaces* command.	N/A

Example

prompt> **nat list globalpools extinterface**
NAT global address pool:
ID | Name | Type | Subnet | IP address | Mask/End Address

```
1   | gp1   |    dmz     | true     | 192.168.102.3 | 255.255.255.0
2   | g2    | internal   | false    | 192.168.103.2 | 192.168.103.50
-------------------------------------------------------------------------------
```

1.198. nat list resvmaps

Syntax

nat list resvmaps <interfacename>

Description

This command lists the following reserved mapping information for a specific outside security interface:

- Reserved mapping identification number
- Reserved mapping name
- Global address - the IP address of the outside interface that is mapped to the inside IP address
- Internal address - the IP address inside the network that the outside security interface IP address is mapped to
- Transport type (IGMP, IPIP etc.)
- Port - TCP or UDP port used by the transport type. If a non- TCP/UDP protocol is used, the port is set to *0*.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
interfacename	The name of an existing security interface (external or DMZ) created and connected to an inside interface (DMZ or internal) using the *nat enable* command. To display security interfaces, use the *security list interfaces* command.	N/A

Example

prompt> **nat list resvmaps extinterface**

```
NAT reserved mappings:
ID | Name | Global Address    | Internal Address    | Type    | Port
-------------------------------------------------------------------------------
1   | rm2   | 192.168.103.2     | 10.10.10.10         |    tcp     | 25
2   | rm1   | 192.168.103.15    | 20.20.20.20         |    udp     | 21
-------------------------------------------------------------------------------
```

1.199. nat show globalpool

Syntax

nat show globalpool <name> <interfacename>

Description

This command displays information about a single network address pool that has been added to an outside interface:

- Type of inside interface (internal or DMZ)
- Subnet configuration status (true if the network pool was set using a subnet mask, false if it was set using a range of IP addresses)

- IP address - the outside network IP address or the first address in the range of addresses
- Subnet Mask or End Address - the subnet mask of the outside network IP address or the last address in the range of addresses

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing global IP address. To display global IP addresses, use the *nat list globalpools* command.	N/A
interfacename	The name of an existing security interface (external or DMZ) created and connected to an inside interface (DMZ or internal) using the *nat enable* command. To display security interfaces, use the *security list interfaces* command.	N/A

Example

prompt> **nat show globalpool gpl extinterface**
NAT global address pool: gp1
Interface type: dmz
Subnet configuration: true
IP address: 192.168.102.3
Subnet mask or End Address: 255.255.255.0

1.200. nat show resvmap

Syntax

nat show resvmap <name> <interfacename>

Description

This command displays the following information about a single reserved mapping configuration that has been added to an outside security interface:

- Global IP address
- Internal IP address
- Transport type
- Port number

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing global IP address. To display global IP addresses, use the *nat list globalpools* command.	N/A
interfacename	The name of an existing security interface (external or DMZ) created and connected to an inside interface (DMZ or internal) using the *nat enable* command. To display security interfaces, use the *security list interfaces* command.	N/A

Example

prompt> **nat show resvmap rm1 extinterface**
NAT reserved mapping: rm1
Global IP address: 192.168.103.15
Internal IP address: 20.20.20.20
Transport type: tcp
Port number: 25

1.201. nat status

Syntax

nat status

Description

This command lists the outside security interfaces and inside interface types that NAT is currently enabled between. It displays the following information:

- NAT object identification number
- NAT object name
- Outside security interface name
- Inside interface type

Example

prompt> **nat status**
NAT enabled on:
ID | Name | Interface | Type
---
1 | n2 | ip2 | internal
2 | n1 | if1 | internal

Port CLI Commands

1.202. port ethernet set

This command is not applicable for the 3648-80 router with the 8 port switch.

Syntax

port ethernet set <attribute> <value>

Description

This command allows you to modify attributes on a ethernet port. Any modifications override existing attribute values specified in your ISOS device and compiled at run-time. The attributes available depend on:

• the type of port that you are using
• the ISOS system that you are using

To display a list of valid attributes for a specific port, use the '*?*' syntax options key after *port Ethernet set*. For example:

prompt> **port ethernet set ?**
100BaseFullAdvert
100BaseHalfAdvert
10BaseFullAdvert
10BaseHalfAdvert
AutoNegotiation
AutoNegotiateRestart
EnableDuplexCheck
Loopback
NoNeg100BaseMode
NoNegFullDuplexMode
PowerDown
Reset

Once you have identified the attribute that you want to modify, you can specify the new value that you want to set it to.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
attribute	A single attribute of the ethernet port. An attribute has a value attached to it which you can modify.	N/A
value	A value attached to an attribute. The value could be a numerical setting or a true/false qualifier.	N/A

Example

prompt> **port ethernet set Loopback true**

1.203. port ethernet show

This command display only 3 items for the 3648-80 router with the 8 port switch.

Syntax

port ethernet show

Description

This command displays the current attributes and values of a ethernet port. The attributes available depend on:

• the type of port that you are using
• the ISOS system that you are using

Example

--> port ethernet show

```
Version                                 = 1.01
100Base                                  = false
100BaseFullAdvert                       = true
100BaseHalfAdvert                        = true
10BaseFullAdvert                        = true
10BaseHalfAdvert                         = true
AutoNegAckOk                               = false
AutoNegDone                               = true
AutoNegotiation                         = true
AutoNegotiateRestart                   = false
Connected                                 = true
DisReconnectCount                       = 2
EnableDuplexCheck                        = true
FullDuplex                              = false
Jabber                                  = false
JabberCount                             = 0
LinkSpeed                                = 100000
Loopback                                 = false
NoNeg100BaseMode                           = false
NoNegFullDuplexMode                      = false
Remote100BTFD                              = false
Remote100BTHD                               = false
Remote10BTFD                               = false
Remote10BTHD                                = false
RemoteFault                             = false
RemoteFaultCount                        = 0
PowerDown                                  = false
Reset                                   = false
```

1.204. port fb set

Syntax

port fb set

Description

This command allows you to modify attributes on a framerelay bridge port. Any modifications override existing attribute values specified in your ISOS device and compiled at run-time. The attributes available depend on:

• the type of port that you are using
• the ISOS system that you are using

To display a list of valid attributes for a specific port, use the '*?*' syntax options key after *port fb set*. For example:

prompt> **port fb set ?**
F_PortStatus
F_ConnStatus
SysMngId
Interface
AutoStart
ManagementType
PortSegmentSize
FullReportCycle
UserMaxErrors
NetMaxErrors
UserErrorWindowSize
NetErrorWindowSize
T391_Value
T392_Value

Note: At present, only the AutoStart and ManagementType attributes have the effect of setting. The other attributes are only for querying the informations.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
attribute	A single attribute of the framerelay bridge port. An attribute has a value attached to it which you can modify.	N/A
value	A value attached to an attribute. The value could be a numerical setting or a true/false qualifier.	N/A

Example

prompt> **port fb set ManagementType LMI_User**

1.205. port fb set ManagementType

Syntax

port fb set ManagementType

Description

This command allows you to modify attributes on a framerelay bridge port LMI management type. Any modifications override existing attribute values specified in your ISOS device and compiled at run-time. The attributes available depend on:

• the type of port that you are using
• the ISOS system that you are using

To display a list of valid attributes for a specific port, use the '*?*' syntax options key after *port fb set ManagementType*. For example:

prompt> **port fb set ManagementType ?**
no_maintenance
933A_Network

933A_User
933A_Both
617D_Network
617D_User
617D_Both
LMI_Network
LMI_User
LMI_Both

Once you have identified the attribute that you want to modify, you can specify the new value that you want to set it to. Note that '993A' refers to the standard ITU-T Q.993 Annex A; '617D' refers to the standard ANSI T1.617 Annex D; 'LMI' refers to the original LMI. Both ends of the Frame Relay link must be configured for the same standard (993A, 617D, LMI). For each standard, there are 'Network', 'User', and 'Both' modes. When two router cards are doing a point-to-point communication, the following condition is not supported: one router card uses the Both mode, the other uses the User mode.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
attribute	A single attribute of the framerelay bridge port LMI management type. An attribute has a value attached to it which you can modify.	N/A
value	A value attached to an attribute. The value could be a numerical setting or a true/false qualifier.	N/A

Example

prompt> **port fb set ManagementType LMI_User**

Notice: **You must save your configuration (see *system config save*) and restart your system (see *system restart*) to apply your ManagementType settings. When LMI management type is used, the connection will start to work (data can be transmitted) about 15 seconds after the system is restarted.**

1.206. port fb show

Syntax

port fb show

Description

This command displays the current attributes and values of a framerelay bridge port. The attributes available depend on:

• the type of port that you are using
• the ISOS system that you are using

Example

--> port fb show

```
Version                                  = 1.00
F_PortStatus                             = 0x00000000
F_ConnStatus                             = 0x00000000
```

SysMngId = 0x00000000
PortClassFrameRelay = true
FpHandle = 0x00415720
Interface = hdlc
AutoStart = false
ManagementType = no_maintenance
PortSegmentSize = 0
FmmHandle = 0x00000000
FmmConnHandle = 0x00000000
FullReportCycle = 6
UserMaxErrors = 3
NetMaxErrors = 3
UserErrorWindowSize = 4
NetErrorWindowSize = 4
T391_Value = 10
T392_Value = 16

1.207. port fr set

Syntax

port fr set

Description

This command allows you to modify attributes on a framerelay router port. Any modifications override existing attribute values specified in your ISOS device and compiled at run-time. The attributes available depend on:

• the type of port that you are using
• the ISOS system that you are using

To display a list of valid attributes for a specific port, use the '*?*' syntax options key after *port fr set*. For example:

prompt> **port fr set ?**
F_PortStatus
F_ConnStatus
SysMngId
Interface
AutoStart
ManagementType
PortSegmentSize
FullReportCycle
UserMaxErrors
NetMaxErrors
UserErrorWindowSize
NetErrorWindowSize
T391_Value
T392_Value

Note: At present, only the AutoStart and ManagementType attributes have the effect of setting. The other attributes are only for querying the informations.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
attribute	A single attribute of the framerelay router port. An attribute has a value attached to it which you can modify.	N/A
value	A value attached to an attribute. The value could be a numerical setting or a true/false qualifier.	N/A

Example

prompt> **port fr set ManagementType LMI_User**

1.208. port fr set ManagementType

Syntax

port fr set ManagementType

Description

This command allows you to modify attributes on a framerelay router port LMI management type. Any modifications override existing attribute values specified in your ISOS device and compiled at run-time. The attributes available depend on:

• the type of port that you are using
• the ISOS system that you are using

To display a list of valid attributes for a specific port, use the *'?'* syntax options key after *port fr set ManagementType*. For example:

prompt> **port fr set ManagementType ?**

no_maintenance
933A_Network
933A_User
933A_Both
617D_Network
617D_User
617D_Both
LMI_Network
LMI_User
LMI_Both

Once you have identified the attribute that you want to modify, you can specify the new value that you want to set it to. Note that '993A' refers to the standard ITU-T Q.993 Annex A; '617D' refers to the standard ANSI T1.617 Annex D; 'LMI' refers to the original LMI. Both ends of the Frame Relay link must be configured for the same standard (993A, 617D, LMI). For each standard, there are 'Network', 'User', and 'Both' modes. When two router cards are doing a point-to-point communication, the following condition is not supported: one router card uses the Both mode, the other uses the User mode.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value

attribute	A single attribute of the framerelay router port LMI management type. An attribute has a value attached to it which you can modify.	N/A
value	A value attached to an attribute. The value could be a numerical setting or a true/false qualifier.	N/A

Example

prompt> **port fr set ManagementType LMI_User**

Notice: **You must save your configuration (see *system config save*) and restart your system (see *system restart*) to apply your ManagementType settings. When LMI management type is used, the connection will start to work (data can be transmitted) about 15 seconds after the system is restarted.**

1.209. port fr show

Syntax

port fr show

Description

This command displays the current attributes and values of a framerelay router port. The attributes available depend on:

• the type of port that you are using
• the ISOS system that you are using

Example

--> port fr show

```
Version                    = 1.00
F_PortStatus               = 0x00000000
F_ConnStatus               = 0x00000000
SysMngId                   = 0x00000000
PortClassFrameRelay        = true
FpHandle                   = 0x00415720
Interface                  = hdlc
AutoStart                  = false
ManagementType             = no_maintenance
PortSegmentSize            = 0
FmmHandle                  = 0x00000000
FmmConnHandle              = 0x00000000
FullReportCycle            = 6
UserMaxErrors              = 3
NetMaxErrors               = 3
UserErrorWindowSize        = 4
NetErrorWindowSize         = 4
T391_Value                 = 10
T392_Value                 = 16
```

1.210. port hdlc set

Syntax

port hdlc set

Description

This command allows you to modify attributes on a hdlc port. Any modifications override existing attribute values specified in your ISOS device and compiled at run-time. The attributes available depend on:

• the type of port that you are using
• the ISOS system that you are using

To display a list of valid attributes for a specific port, use the *'?'* syntax options key after *port hdlc set*. For example:

prompt> **port hdlc set ?**

MaxQueue
Reset
Disable
TxClockInvert
RxClockInvert
Loopback
Resync
EnableFlags
ResyncDelay
ClockSpeed

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
attribute	A single attribute of the hdlc port. An attribute has a value attached to it which you can modify.	N/A
value	A value attached to an attribute. The value could be a numerical setting or a true/false qualifier.	N/A

Example

prompt> **port hdlc set ClockSpeed 1000**

Notice: **You must save configuration (see *system config save*) and restart the system (see *system restart*) to apply ClockSpeed settings. The value of ClockSpeed is recommend to be the same with the hdlc clock rate of the primary T1/E1 card. The difference between the clock rate of router card and primary T1/E1 card may result in packet loss.**

1.211. port hdlc show

Syntax

port fb show

Description

This command displays the current attributes and values of a framerelay bridge port. The attributes available depend on:

• the type of port that you are using
• the ISOS system that you are using

Example

--> port hdlc show

```
Version                                  = 1.07
MaxQueue                                 = 32
Connected                                = false
PortClassHDLC                            = true
Reset                                    = false
Disable                                  = false
TxClockInvert                            = false
RxClockInvert                            = false
Loopback                                 = false
Resync                                   = false
EnableFlags                              = true
ResyncDelay                              = 1000
ResyncCntDataOv                          = 0
ResyncCntNoData                          = 0
ResyncCntOrun                            = 0
ClockSpeed                               = 1000
```

PPPoH CLI commands

This chapter describes the PPP over High-Level Data Link Control (HDLC) CLI commands.

1.212. pppoh add transport dialin

Syntax

pppoh add transport <name> dialin <interface> <port>

Description

This command creates a PPPoH transport that accepts dialin connections. It allows you to specify the following information:

- the PPP interface to the channel
- the HDLC port that will transport data

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the transport. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A
interface	The PPP interface to a channel that transports PPPoH data. A single interface can be used by multiple channels. The interface value can be any positive integer.	N/A
port	The system port that is used to transport HDLC data.	N/A

Example

prompt> **pppoh add transport pppoh1 dialin 1 hdlc**

1.213. pppoh add transport dialout

Syntax

pppoh add transport <name> dialout <interface> <port>

Description

This command creates a PPPoH transport that performs dialout. It allows you to specify the following information:

- the PPP interface to the channel
- the HDLC port that will transport data

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the transport. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A
interface	The PPP interface to a channel that transports PPPoH data. A single interface can be used by multiple channels. The interface value can be any positive integer.	N/A
port	The system port that is used to transport HDLC data.	N/A

Example

prompt> **pppoh add transport pppoh1 dialout 1 hdlc**

1.214. pppoh clear transports

Syntax

pppoh clear transports

Description

This command deletes all PPPoH transports that were created using the *pppoh add transport* commands.

Example

prompt> **pppoh clear transports**

1.215. pppoh delete transport

Syntax

pppoh delete transport {<name>|<number>}

Description

This command deletes a single PPPoH transport.

Options

The following table gives the range of values for each option which can be specified with this command and a default value for each option (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport numbers, use the *pppoh list transports* command.	N/A

Example

prompt> **pppoh delete transport pppoh1**

1.216. pppoh list transports

Syntax

pppoh list transports

Description

This command lists PPPoH transports that have been created using the *pppoh add transport* commands. It displays the following information about the transports:

- transport identification number
- transport name

Example

prompt> **pppoh list transports**
PPPOH transports:
ID	Name
1	p2
2	p1

1.217. pppoh set transport createroute

Syntax

pppoh set transport {<name>|<number>} createroute {enabled|disabled}

Description

This command specifies whether a route is added to the system after IPCP (Internet Protocol Control Protocol) negotiation is completed. If set to *enabled*, a route will be created which directs packets to the remote end of the PPP link.This route can either be a default route or a specific route, depending on the value set using the *pppoa set transport specificroute* command. To display the *createroute* setting, use the *pppoh show transport* command. The route is removed when the PPP link is disconnected.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
enabled	Adds a route to the system after IPCP negotiation.	enabled
disabled	Does not add a route to the system after IPCP negotiation.	

Example

prompt> **pppoh set transport pppoh1 createroute disabled**

1.218. pppoh set transport dialin

Syntax

pppoh set transport {<name>|<number>} dialin

Description

This command sets an existing PPPoH transport to accept dialin connections. This replaces the transports existing dialin/dialout setting. The transport uses the interface that was specified when the transport was created.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A

Example

prompt> **pppoh set transport pppoh2 dialin**

1.219. pppoh set transport dialout

Syntax

pppoh set transport {<name>|<number>} dialout

Description

This command sets a PPPoH transport to perform dialout. This replaces the transports existing dialin/dialout setting. The transport uses the interface that was specified when the transport was created.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A

Example

prompt> **pppoh set transport pppoh2 dialout**

1.220. pppoh set transport discoverdns primary

Syntax

pppoh set transport {<name>|<number>} discoverdns primary {enabled|disabled}

Description

This command enables/disables whether the primary DNS server address is requested from a remote PPP peer using IPCP. The default setting for this command is enabled. The default setting for the *pppoh set transport givedns* commands is also enabled.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
enabled	A primary DNS server IP address is requested.	enabled
disabled	A primary DNS server IP address is not requested.	

Example

prompt> **pppoh set transport pppoh3 discoverdns primary enabled**

1.221. pppoh set transport discoverdns secondary

Syntax

pppoh set transport {<name>|<number>} discoverdns secondary {enabled|disabled}

Description

This command enables/disables whether the secondary DNS server address is requested from a remote PPP peer using IPCP. The default setting for this command is enabled. The default setting for the *pppoh set transport givedns* commands is also enabled.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
enabled	A primary DNS server IP address is requested.	enabled
disabled	A primary DNS server IP address is not requested.	

Example

prompt> **pppoh set transport pppoh3 discoverdns secondary enabled**

1.222. pppoh set transport enabled|disabled

Syntax

pppoh set transport {<name>|<number>} {enabled|disabled}

Description

This command explicitly enables/disables a PPPoH transport. Attaching a transport to an interface implicitly enables it, but for cases where no attach is performed (for example, multiple channels on an interface, a PPP session that is not attached but needed for testing purposes) the transport must be enabled explicitly.

Options

The following table gives the range of values for each option which can be specified with this command and a default value for each option (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
enabled	Enables a PPPoH transport.	enabled
disabled	Disables a PPPoH transport.	

Example

prompt> **pppoh set transport pppoh1 enabled**

1.223. pppoh set transport givedns client enabled|disabled

Syntax

pppoh set transport {<name>|<number>} givedns client {enabled | disabled}

Description

This command controls whether the PPP Internet Protocol Control Protocol (IPCP) can request a DNS server IP address for a remote PPP peer. Once IPCP has discovered the DNS server IP address, it automatically gives the address to the local DNS client so that a connection can be established. You must have the DNS client process included in your image build in order to use this protocol.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
enabled	IPCP can request a DNS server IP address and then give the address to DNS client.	enabled
disabled	IPCP cannot request a DNS server IP address and then give the address to DNS client.	

Example

prompt> **pppoh set transport pppoh1 givedns client enabled**

1.224. pppoh set transport givedns relay enabled|disabled

Syntax

pppoh set transport {<name>|<number>} givedns relay {enabled | disabled}

Description

This command controls whether the PPP Internet Protocol Control Protocol (IPCP) can request the DNS server IP address for a remote PPP peer. Once IPCP has discovered the DNS server IP address, it automatically gives the address to the local DNS relay so that a connection can be established. You must have the DNS relay process included in your image build in order to use this protocol.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
enabled	IPCP can request a DNS server IP address and then give the address to DNS relay.	enabled
disabled	IPCP cannot request a DNS server IP address and then give the address to DNS relay.	

Example

prompt> **pppoh set transport pppoh1 givedns relay enabled**

1.225. pppoh set transport headers hdlc

Syntax

pppoh set transport {<name>|<number>} headers hdlc {enabled|disabled}

Description

This command allows you to enable/disable whether your system can transmit and receive packets containing HDLC headers. HDLC headers should **always** be enabled - if you disable them using this command, you will not be able to transport any HDLC packets. If you want LLC packets to be transmitted and received as well as HDLC packets, use the *pppoh set transport headers llc enable* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
enabled	Packets that have HDLC headers can be transmitted/received.	enabled
disabled	Packets that have HDLC headers can not be transmitted/received.	

Example

prompt> **pppoh set transport pppoh1 headers hdlc enabled**

1.226. pppoh set transport headers llc

Syntax

pppoh set transport {<name>|<number>} headers llc {enabled|disabled}

Description

This command allows you to enable/disable whether your system can transmit and receive packets containing LLC headers. By default, HDLC packets are **always** transmitted and received. See the *pppoh set transport headers hdlc enable* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list*	N/A

	transports command.	
enabled	Packets that have LLC headers can be transmitted/received.	enabled
disabled	Packets that have LLC headers can not be transmitted/received.	

Example

prompt> **pppoh set transport pppoh1 headers llc enabled**

1.227. pppoh set transport interface

Syntax

pppoh set transport {<name>|<number>} interface <interface>

Description

This command sets the PPP interface for an existing PPPoH transport.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
interface	The PPP interface to a channel that transports PPPoH data. A single interface can be used by multiple channels. The interface value can be any positive integer.	N/A

Example

prompt> **pppoh set transport pppoh2 interface 4**

1.228. pppoh set transport lcpechoevery

Syntax

pppoh set transport {<name>|<number>} lcpechoevery <interval>

Description

This command tells a specified PPP transport to send an LCP (Link Control Protocol) echo request frame at specified intervals (in seconds). If no reply to the request is received, the PPP connection is torn down. This functionality is also known as 'keep-alive'. If you do not want to send LCP echo frames, specify zero (0) in the <interval> attribute.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport.	N/A

	To display transport names, use the *pppoh list transports* command.	
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
interval	The length of time (in seconds) between LCP echo request frames being sent. If you do not want echo request frames to be sent, specify '0' as the interval.	10 seconds

Example

prompt> pppoh set transport pppoh2 lcpechoevery 0

1.229. pppoh set transport lcpmaxconf

Syntax

pppoh set transport {<name>|<number>} lcpmaxconf <lcp max configure>

Description

This command sets the Link Control Protocol (LCP) maximum parameter for an existing PPPoH transport.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
lcp max configure	Link Control Protocol; the maximum number of configures that can be transmitted without reply before assuming that the destination address is unable to respond. The LCPmaxconf can be any positive value.	10

Example

prompt> **pppoh set transport pppoh1 lcpmaxconf 20**

1.230. pppoh set transport lcpmaxfail

Syntax

pppoh set transport {<name>|<number>} lcpmaxfail <lcp max fail>

Description

This command sets the Link Control Protocol (LCP) maximum fail parameter for an existing PPPoH transport.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
lcp max fail	Link Control Protocol; the maximum number of consecutive negative acknowledgements (indicating that the information received contains errors) that can be transmitted before assuming that parameter negotiation is not converging. The LCPmaxfail can be any positive value.	5

Example

prompt> **pppoh set transport pppoh1 lcpmaxfail 20**

1.231. pppoh set transport lcpmaxterm

Syntax

pppoh set transport {<name>|<number>} lcpmaxterm <lcp max terminate>

Description

This command sets the Link Control Protocol (LCP) maximum terminate parameter for an existing PPPoH transport.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
cp max term	Link Control Protocol; the maximum number of consecutive Terminate Requests that will be sent without reply before assuming that the destination address is unable to respond. The LCPfailterm can be any positive value.	2

Example

prompt> **pppoh set transport pppoh1 lcpmaxterm 20**

1.232. pppoh set transport localip

Syntax

pppoh set transport {<name>|<number>} localip <ip-address>

Description

This command is only applicable to dialin transports that provide the server-end of a connection. The

command tells the PPP process the local IP address to be associated with the client-end of an interface. This allows remote users to have dialin access via the channel(s) that the interface is attached to.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
ip-address	The IP address of the local 'client-end' of an interface displayed in the following format: 111.222.254.4	0.0.0.0

Example

prompt> **pppoh set transport pppoh1 localip 192.168.103.2**

1.233. pppoh set transport password

Syntax

pppoh set transport {<name>|<number>} password <password>

Description

This command sets a dial-out password on a named transport. The password is required when PPP negotiation takes place and is supplied to the remote PPP server for authentication.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
password	An arbitrary word that acts as a dialout password enabling you to login to the remote end. The password will be required by the PPP server when you want to login remotely. It can be made up of one or more characters and/or digits. To display the password, use the *pppoh show transport* command.	N/A

Example

prompt> **pppoh set transport pppoh2 password mercury**

1.234. pppoh set transport remotedns

Syntax

pppoh set transport {<name>|<number>} remotedns <ipaddress> [<ipaddress2>]

Description

This command is a *PPP server* function. This command sets the primary and secondary local DNS server addresses that will be given to a remote PPP peer when the peer requests a primary or secondary DNS server IP address using IPCP. Setting the secondary IP address is optional. If you want to delete an IP address, set the IP address to *0.0.0.0*.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
ipaddress	The ip address of the primary local DNS server displayed in the following format: 192.168.102.3	0.0.0.0 (no primary address set)
ipaddress2	The ip address of the secondary local DNS server displayed in the following format: 192.168.102.3	0.0.0.0 (no secondary address set)

Examples

Example One - setting a primary address

prompt> **pppoh set transport pppoh1 remotedns 192.168.102.3**

Example Two - setting primary and secondary addresses

To set primary and secondary addresses, use this command syntax:

prompt> **pppoh set transport pppoh1 remotedns 192.168.102.3 192.168.105.1**

Example Three - deleting an address

To delete an address, set it to *0.0.0.0*. The example below deletes the secondary address that was set in Example Two:

prompt> **pppoh set transport pppoh1 remotedns 192.168.102.3 0.0.0.0**

1.235. pppoh set transport remoteip

Syntax

pppoh set transport {<name>|<number>} remoteip <ip-address>

Description

This command sets the IP address supplied to the remote end of the PPP connection during negotiation. This is particularly important for PPP dialin transports. If the remote peer doesn't set its IP address for PPP connection, it will use the IP set in this field. But if the remote peer already set its IP address for PPP connection, you must not set the Remote IP or the connection can't be established.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value

name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
Ip-address	The IP address of the local 'server-end' of an interface displayed in the following format: 192.168.102.3	0.0.0.0

Example

prompt> **pppoh set transport pppoh1 remoteip 192.168.103.2**

1.236. pppoh set transport routemask

Syntax

pppoh set transport {<name>|<number>} routemask <mask>

Description

This command sets the subnet mask used by the route that is created when a PPP link comes up. If it is set to *0.0.0.0*, the subnet mask is determined by the IP address of the remote end of the link. The class of the IP address is obtained during IPCP (Internet Protocol Control Protocol) negotiation.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
mask	The subnet mask that is used for the route that is created when a PPP link comes up. 0.0.0.0	0.0.0.0

Example

prompt> **pppoh set transport pppoh1 routemask 0.0.0.0**

1.237. pppoh set transport specificroute

Syntax

pppoh set transport {<name>|<number>} specificroute {enabled | disabled}

Description

This command specifies whether the route created when a PPP link comes up is a specific or default route. If set to *enabled*, the route created will only apply to packets for the subnet at the remote end of the PPP link. The address of this subnet is obtained during IPCP negotiation. The mask for the route is calculated from the class of the remote subnet unless an alternative has been specified using the *pppoh set transport routemask* command. If specificroute is set to *disabled*, a default route to the subnet at the remote end of the PPP link is created. Note that the current setting of this command is ignored if *pppoh set transport createroute* command is set to *disabled*.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
enabled	Allows the created route to apply to packets for the subnet at the remote end of the PPP link.	disabled
disabled	A default route to the subnet at the remote end of the PPP link is created.	

Example

prompt> **pppoh set transport pppoh1 specificroute disabled**

1.238. pppoh set transport subnetmask

Syntax

pppoh set transport {<name>|<number>} subnetmask <mask>

Description

This command sets the subnet mask used for the local IP interface connected to the PPP transport. If the value *0.0.0.0* is supplied, the netmask will be calculated from the class of the IP address obtained during IPCP negotiation.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
mask	The subnet mask that is used for the route that is created when a PPP link comes up. 0.0.0.0	0.0.0.0

Example

prompt> **pppoh set transport pppoh1 subnetmask 255.255.255.0**

1.239. pppoh set transport theylogin

Syntax

pppoh set transport {<name>|<number>} theylogin {none|pap|chap}

Description

This command sets the authentication method that remote PPP clients must use to dialin to the router. If authentication is used, clients must use the specified authentication method and provide the username set

using the *system add user* command. This command is only valid if the user has maydialin set using the *system set login maydialin* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
none	No authentication method is set.	None
pap	Password Authentication Protocol; the server sends an authentication request to the remote user dialing in. PAP passes the *unencrypted* username and password and identifies the remote end.	
chap	Challenge Handshake Authentication Protocol; the server sends an authentication request to the remote user dialing in. PAP passes the *encrypted* username and password and identifies the remote end.	

Example

prompt> **pppoh set transport pppoh2 theylogin pap**

1.240. pppoh set transport username

Syntax

pppoh set transport {<name>|<number>} username <username>

Description

This command sets a (dial-out) username on a named transport. The username is required when PPP negotiation takes place and is supplied to the remote PPP server for authentication.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
username	A name that identifies a user and, together with the dialout password, enables a user to login to the remote end. The username will be required by the PPP server when the user wants to login remotely. It can be made up of one or more characters and/or	N/A

	digits. To display the username, use the *pppoh show transport* command.	

Example

prompt> **pppoh set transport pppoh2 username jsmith**

1.241. pppoh set transport welogin

Syntax

pppoh set transport {<name>|<number>} welogin {none| pap|chap}

Description

This command sets the authentication protocol used to connect to external PPP servers (dial-out).

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
none	No authentication method is set.	None
pap	Password Authentication Protocol; the server sends an authentication request to the remote user dialing in. PAP passes the *unencrypted* username and password and identifies the remote end.	
chap	Challenge Handshake Authentication Protocol; the server sends an authentication request to the remote user dialing in. PAP passes the *encrypted* username and password and identifies the remote end.	

Example

prompt> **pppoh set transport pppoh2 welogin pap**

1.242. pppoh show transport

Syntax

pppoh show transport {<name>|<number>}

Description

This command displays the following information about an existing PPPoH transport:

• Description
• Summary - the connection state
• Server - dialin status
• HDLC header status - whether the transport can accept or receive packets in HDLC data format (true or false). This should always be true for a PPPoH transport.
• LLC header status - whether the transport can accept or receive

packets in LLC data format (true or false).
• Local IP address
• Subnet mask
• Remote IP address
• Remote DNS status
• Give DNS to Client status
• Give DNS to Relay status
• Create route status
• Specific route status
• Route mask
• Dialout Username
• Dialout Password
• Dialout Authentication method
• Dialin Authentication method
• LCP Max Configure
• LCP Max Failure
• LCP Max Terminate
• LCP Echo Every

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A
number	A number that identifies an existing PPPoH transport. To display transport names, use the *pppoh list transports* command.	N/A

Example

prompt> **pppoh show transport h1**
PPP Transport: h1
Description : h1
Summary : disabled
Server : false
HDLC : true
LLC : false
Local Ip : 0.0.0.0
Subnet Mask : 0.0.0.0
Remote Ip : 0.0.0.0
Remote DNS : N/A
Give DNSto Client : true
Give DNSto Relay : true
Create Route : true
Specific Route : false
Route Mask : 0.0.0.0
Dialout Username :
Dialout Password :
Dialout Auth : none

Dialin Auth : none
Lcp Max Configure : 10
Lcp Max Failure : 5
Lcp Max Terminate : 2
Lcp Echo Every : 10

PPTP CLI commands

1.243. anscpptp set pool

Syntax

anscpptp set pool <pool_start_addr> <pool_end_addr>

Description

This command allows you to set the starting and ending IP address of the pptp ip pool.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
pool_start_addr	Starting IP Address of the pptp ip pool.	N/A
pool_end_addr	Ending IP Address of the pptp ip pool.	N/A

Example

anscpptp set pool 172.168.2.128 172.168.2.191

1.244. anscpptp show pool

Syntax

anscpptp show pool

Description

This command shows the IP pool of ansc pptp.

Example

--> anscpptp show pool
Ip Pool Start: 172.168.2.128
Ip Pool End: 172.168.2.191

1.245. anscpptp show client

Syntax

anscpptp show client

Description

This command shows connected clients.

Example

--> anscpptp show client
Client List:

Security CLI commands
This chapter describes the Security CLI commands.

1.246. security add interface

Syntax
security add interface <name> {external|internal|dmz}

Description
This command adds an existing IP interface to the Security package to create a security interface, and specifies what type of interface it is depending on how it connects to the network. Once you have added security interfaces, you can use them in the NAT and/or Firewall configurations.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
external	An interface that connects to the external network.	N/A
internal	An interface that connects to the internal network	N/A
dmz	An interface that connects to the demilitarized zone (DMZ)	N/A

Example
prompt> **security add interface ip1 internal**

1.247. security add trigger netmeeting

Syntax
security add trigger <name> netmeeting

Description
This command allows you to use the example trigger provided by the CLI. It allows you to add a trigger to allow Netmeeting to transport data through the security package. This application opens a secondary port session. You do not have to set the port range or *maxactinterval* for a Netmeeting trigger - the CLI automatically sets this for you.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the trigger. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A

Example
prompt> **security add trigger t2 netmeeting**

1.248. security add trigger tcp|udp

Syntax

security add trigger <name> {tcp|udp} <startport> <endport> <maxactinterval>

Description

This command adds a trigger to the Security module. A trigger allows an application to open a secondary port in order to transport packets. Some applications, such as FTP, need to open secondary ports – they have a control session port (21 for FTP) but also need to use a second port in order to transport data. Adding a trigger means that you do not have to define static portfilters to open ports for each secondary session. If you did this, the ports would remain open for potential use (or misuse, see the command *firewall set IDS SCANattackblock*) until the portfilters were deleted. A trigger opens a secondary port dynamically, and allows you to specify the length of time that it can remain inactive before it is closed.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the trigger. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A
tcp	Adds a trigger for a TCP application to the security package.	N/A
udp	Adds a trigger for a UDP application to the security package.	N/A
startport	Sets the start of the trigger port range for the control session.	N/A

Example

The following example creates a Netmeeting (H323) trigger:
prompt> **security add trigger t1 tcp 1720 1720 30000**

1.249. security clear interfaces

Syntax

security clear interfaces

Description

This command removes all security interfaces that were added to the Security package using the *security add interface* command.

Example

prompt> **security clear interfaces**

1.250. security clear triggers

Syntax

security clear triggers

Description

This command deletes all triggers that were added to the Security module using the *security add trigger* commands.

Example

prompt> **security clear triggers**

1.251. security delete interface

Syntax

security delete interface <name>

Description

This command removes a single security interface that was added to the Security package using the *security add interface* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A

Example

prompt> **security delete interface f1**

1.252. security delete trigger

Syntax

security delete trigger <name>

Description

This command deletes a single trigger that was added to the Security module using the *security add trigger* commands.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing trigger. To display trigger names, use the *security list trigger* command.	N/A

Example

prompt> **security delete trigger t2**

1.253. security enable|disable

Syntax

security {enable | disable}

Description

This command explicitly enables/disables all modules in the Security package (including the child modules; NAT and Firewall). You **must** enable the Security package if you want to use the NAT and/or Firewall modules to configure security for your system. If you disable the Security package during a session, any configuration changes made to the Security, NAT or Firewall modules when the package was enabled remain in the system, so that you can re-enable them later in the session. If you need to reboot

your system but want to save the security configuration between sessions, use the *system config save* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enabled	Enables all modules in the Security package (Security, NAT and Firewall modules).	disabled
disabled	Disables all modules in the Security package (Security, NAT and Firewall modules).	

Example

prompt> **security enable**

1.254. security list interfaces

Syntax

security list interfaces

Description

This command lists the following information about security interfaces that were added to the Security package using the *security add interface* command:

- Interface ID number
- Interface name
- Interface type (external, internal or DMZ)

Example

prompt> **security list interfaces**

```
Security Interfaces:
ID | Name    | Type
---|---------|---------
1  | i1      | internal
2  | i2      | external
3  | i3      | dmz
--------------------------
```

1.255. security list triggers

Syntax

security list triggers

Description

This command lists triggers that were added to the Security module using the *security add trigger* command. It displays the following information about triggers:

- Trigger ID number
- Trigger name
- Trigger transport type (TCP or UDP)
- Port range
- Interval

Example

prompt> **security list triggers**

Security Triggers:
ID | Name | Type | Port Range | Interval
---
1 | tr1 | tcp | 21 – 21 | 3000
2 | tr2 | tcp | 1720 - 1720 | 3000

1.256. security set trigger addressreplacement

Syntax

security set trigger <name> addressreplacement {none|tcp|udp|both}

Description

The settings in this command are only effective if you enable address translation using the command *security set trigger binaryaddressreplacement*. This command allows you to specify what type of address replacement is set on an trigger. Incoming packets are searched in order to find their embedded IP address. The address is then replaced by the correct inside host IP address, and NAT translates the packets to the correct destination. You can specify whether you want to carry out address replacement on TCP packets, on UDP packets or on both TCP and UDP packets.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an trigger. To display trigger names, use the *security list triggers* command.	N/A
none	Disables address replacement.	none
tcp	Sets address replacement on TCP packets for an existing trigger.	
udp	Sets address replacement on UDP packets for an existing trigger.	
both	Sets address replacement on TCP and UDP packets for an existing trigger.	

Example

prompt> **security set trigger t2 addressreplacement tcp**

1.257. security set trigger binaryaddressreplacement

Syntax

security set trigger <name> binaryaddressreplacement {enable | disable}

Description

This command enables/disables binary address replacement on an existing trigger. You can then set the type of address replacement (TCP, UDP, both or none) using the command *security set trigger addressreplacement*.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an trigger. To display trigger names, use the *security list triggers* command.	N/A
enable	Enables the use of binary address replacement on an existing trigger.	disable
disable	Disables the use of binary address replacement on an existing trigger.	

Example

prompt> **security set trigger t5 binaryaddressreplacement enable**

1.258. security set trigger endport

Syntax

security set trigger <name> endport <portnumber>

Description

This command sets the end of the port number range for an existing trigger.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an trigger. To display trigger names, use the *security list triggers* command.	N/A
portnumber	Sets the end of the trigger port range.	N/A

Example

prompt> **security set trigger t3 endport 21**

1.259. security set trigger maxactinterval

Syntax

security set trigger <name> maxactinterval <interval>

Description

This command sets the maximum activity interval limit on existing session entries for an existing trigger.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an trigger. To display trigger names, use the *security list triggers* command.	N/A

interval	Sets the maximum interval time (in milliseconds) between the use of secondary port sessions. If a secondary port opened by a trigger has not been used for the specified time, it is closed.	N/A

Example

prompt> **security set trigger t2 maxactinterval 5000**

1.260. security set trigger multihost

Syntax

security set trigger <name> multihost {enable | disable}

Description

This command sets whether or not a secondary session can be initiated to/from different remote hosts or the same remote host on an existing trigger.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an trigger. To display trigger names, use the *security list triggers* command.	N/A
enable	A secondary session can be initiated to/from different remote hosts.	disable
disable	A secondary session can only be initiated to/from the same remote host.	

Example

prompt> **security set trigger t1 multihost enable**

1.261. security set trigger sessionchaining

Syntax

security set trigger <name> sessionchaining {enable | disable}

Description

This command determines whether or not a triggering protocol can be chained. If session chaining is enabled, TCP dynamic sessions also become triggering sessions, which allows multi-level session triggering.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an trigger. To display trigger names, use the *security list triggers* command.	N/A
enable	Enables TCP sessionchaining on an existing trigger.	disable

disable	Disables all session chaining (TCP and UDP) on an existing trigger.	

Example

prompt> **security set trigger t4 sessionnchaining enable**

1.262. security set trigger startport

Syntax

security policy <name> set trigger startport <portnumber>

Description

This command sets the start of the port number range for an existing trigger.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an trigger. To display trigger names, use the *security list triggers* command.	N/A
portnumber	Sets the start of the trigger port range.	N/A

Example

prompt> **security set trigger t3 startport 21**

1.263. security set trigger UDPsessionchaining

Syntax

security set trigger <name> UDPsessionchaining {enable | disable}

Description

You **must** set the *security set trigger sessionchaining enable* command in order for this command to become effective. If UDP session chaining is enabled, both UDP and TCP dynamic sessions also become triggering sessions, which allows multi-level session triggering.

Note - This CLI command is **case-sensitive**. You *must* type the command attributes exactly as they appear in the syntax section on this page. If you do not use the same case-sensitive syntax, the command fails and the CLI displays a syntax error message.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an trigger. To display trigger names, use the *security list triggers* command.	N/A
enable	Enables UDP sessionchaining on an existing trigger. TCP and UDP session chaining is allowed if the *security set trigger sessionchaining* command is enabled.	disable

disable	Disables UDP session chaining on an existing trigger. TCP session chaining is allowed if the *security set trigger sessionchaining* command is enabled.	

Example

prompt> **security set trigger t3 UDPsessionchaining enable**

1.264. security show interface

Syntax

security show interface <name>

Description

This command displays information about a single interface that was added to the Security package using the *security add interface* command. The following interface information is displayed:

- Interface name
- Interface type (external, internal or DMZ)

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing security interface. To display all Interface names, use the *security list interfaces* command.	N/A

Example

prompt> **security show interface f2**
Interface name: f2
Interface type: internal

1.265. security show trigger

Syntax

security show trigger <name>

Description

This command displays information about a single trigger that was added to the Security module using the *security add trigger* command.
The following trigger information is displayed:

- Trigger name
- Transport type (TCP or UDP)
- Start of the port range
- End of the port range
- Multiple host permission (true/false)
- Maximum activity interval (in milliseconds)
- Session chaining permission (true/false)
- Session chaining on UDP permission (true/false)
- Binary address replacement permission (true/false)
- Address translation type (UDP, TCP, none or both)

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an trigger. To display trigger names, use the *security list triggers* command.	N/A

Example

prompt> **security show trigger t2**
Security Trigger: t2
Transport Type: tcp
Starting port number: 1000
Ending port number: 1000
Allow multiple hosts: false
Max activity interval: 30000
Session chaining: false
Session chaining on UDP: false
Binary address replacement: false
Address translation type: none

1.266. security status

Syntax

security status

Description

This command displays the following information about the Security package:

- Security status (enabled or disabled)
- Firewall status (enabled or disabled)
- Firewall security level setting (none, high, low, or medium)
- Firewall session logging (enabled or disabled)
- Firewall blocking logging (enabled or disabled)
- Firewall intrusion logging (enabled or disabled)
- NAT status (enabled or disabled)

Example

prompt> **security status**
Security enabled.
Firewall disabled.
Firewall security level: none.
Firewall session logging enabled.
Firewall blocking logging enabled.
Firewall intrusion logging disabled.
NAT enabled

SNMP CLI commands

1.267. snmp add community

Syntax

snmp add community <commstr> {v1|v2c} [hostname<hostname>] [rw]

Description

This command allows you to add SNMP new community information including the community string(name), the version of SNMP(v1 or v2c), the host name under the community, and the access right.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
commstr	Required, to specify the community string for SNMPv1 or SNMPv2c.	N/A
hostname	Optional, to specify host name of this SNMP community . Validation: Host name must exist at SNMP Host Table.	N/A

Example

snmp add community test v1 rw

1.268. snmp add host

Syntax

snmp add host <hostname> <ipaddr> [port <ipport>] [mask <mask>] [{v1|v2c} <commstr>]

Description

This command allows you to add SNMP new host information including the hostname, the IP address of the host, the port and mask of the host, the SNMP version, and the community string.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
hostname	To specify host name of this SNMP community. Validation: Host name must exist at SNMP Host Table.	N/A
ipaddr	IP address.	N/A
ipport	Optional, Port..	N/A
mask	Optional, Netmask.	N/A
commstr	Optional, Community string.	N/A

Example

snmp add host h1 192.168.2.46 v1 test

1.269. snmp add trap

Syntax

snmp add trap <trapname> <hostname>

Description

This command allows you to add SNMP new trap information. There are four types of traps: Cold start, CGA trap, Loss of the Ethernet data link, SNMP Authentication trap.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
trapname	Name of this SNMP trap.	N/A
hostname	Host name. Validation: Access list name must exist at SNMP Host Table.	N/A

Example

snmp add trap t1 h1

1.270. snmp config save

Syntax

snmp config save

Description

This command allows you to save the SNMP configuration information.

Example

snmp config save

1.271. snmp delete community

Syntax

snmp delete community <commstr>

Description

This command allows you to delete the specified SNMP community information.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
Commstr	Community string	N/A

Example

snmp delete community test

1.272. snmp delete host

Syntax

snmp delete host <hostname>

Description

This command allows you to delete the specified SNMP host destination information.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
hostname	Host name	N/A

Example

snmp delete host h1

1.273. snmp delete trap

Syntax

snmp delete trap <trapname>

Description

This command allows you to delete the specified SNMP trap information.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
trapname	trap name	N/A

Example

snmp delete trap t1

1.274. snmp show community

Syntax

snmp show community

Description

This command allows you to display SNMP community information.

Example

snmp show community

```
Community Name   Group Name     Read View    Write View   Security Model   Host Name        IP Address       Mask
-------------------------- ---------------- ------------ ---------- ---------------------- ---------------- ----------

public           V1RWGroup      All          All          SNMPv1           -                -                -
```

1.275. snmp show host

Syntax

snmp show host

Description

This command allows you to display SNMP host destination information.

Example

snmp show host

Host Name	IP:Port	Mask	Trap Community	Trap version	Tag List
v1Target	192.168.7.21:0	255.255.0.0	public	SNMPv1	TrapReceiver

1.276. snmp show trap

Syntax

snmp show trap

Description

This command allows you to display SNMP trap destination information.

Example

snmp show trap

Trap Name	Host Name	Address:Port	Community String	Security Model
dslamNotifyEntry	v1Target	192.168.7.21:0	public	SNMPv1

SNTP CLI commands
This chapter describes the SNTP (Simple Network Time Protocol) CLI commands.

1.277. sntpclient set clock

Syntax
sntpclient set clock <yyyy:mm:dd:hh:mm:ss>

Description
This command sets the ISOS system clock to a specific time and date. This command can be used as an alternative to synchronizing the local system clock via internal or external timeservers.

Example
The following command sets the ISOS system clock to 11:10:13pm, 2nd November 2001:
prompt> **sntpclient set clock 2001:11:02:23:10:13**

1.278. sntpclient set mode

Syntax
sntpclient set mode {unicast|broadcast|anycast} {enable|disable}

Description
This command enables/disables the STNP client in a particular time synchronous access mode. There are three modes to choose from, and each mode has enable and disable options:

- **Unicast** mode
  - *Enable* - the mode uses a unicast server and the IP address or hostname in the SNTP server association list is used to synchronize the client time with the server. The SNTP client attempts to contact the specific server in the association in order to receive a timestamp when the *sntpclient sync* command is issued.
  - *Disable* - the unicast server is removed from the association list.
- **Broadcast** mode
  - *Enable* - allows the SNTP client to accept time synchronization broadcast packets from an SNTP server located on the network, and updated the local system time accordingly.
  - *Disable* - stops synchronization via broadcast mode
- **Anycast** mode
  - *Enable* - the SNTP client sends time synchronized broadcast packets to the network and subsequently expects a reply from a valid timeserver. The client then uses the first reply it receives to establish a link for future sync operations in unicast mode. This server will then be added to the server association list. The client ignores any later replies from servers after the first one is received. The enabled anycast mode takes precedence over any entries currently in the associations list when the sntpclient sync command is issued. The entry will then be substituted for any existing entry in the unicast association list.
  - *Disable* - stops synchronization via anycast mode.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
unicast	Sets the time synchronous access mode to use the unicast server.	N/A
broadcast	Sets the time synchronous access mode to use the broadcast server.	N/A

anycast	Sets the time synchronous access mode to use the anycast server.	N/A
enable	Enables the selected time synchronous access mode.	N/A
disable	Disables the selected time synchronous access mode.	N/A

Example

prompt> **sntpclient set mode anycast enable**

1.279. sntpclient set poll-interval

Syntax

sntpclient set poll-interval <0-30>

Description

This command sets the SNTP client to automatically send a time synchronization request (specific to the mode) to the network at a specific interval. If the poll-interval is set to 0, the polling mechanism will be disabled.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
0-30	Sets the polling interval (in minutes) that SNTP client will sync with a designated server. This can be any value between 0 and 30.	0(disable)

Example

prompt> **sntpclient set poll-interval 10**

1.280. sntpclient set retries

Syntax

sntpclient set retries <0-10>

Description

This command sets the number of packet retry attempts when no response is received from a timeserver. The SNTP client will send another packet for synchronization after a timeout.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
0-10	Sets the number of packet retry attempts made when no response is received from a timeserver.	2

Example

prompt> **sntpclient set retries 4**

1.281. sntpclient set server

Syntax

sntpclient set server {ipaddress <IP address> | hostname <hostname>}

Description

This command sets the dedicated unicast server for which the SNTP client can synchronize its time. You can set the server either by specifying the IP address or the hostname.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
IP address	The IP address of the dedicated unicast server that SNTP can use to synchronize its time.	N/A
hostname	The hostname of the dedicated unicast server that SNTP can use to synchronize its time.	N/A

Examples

Example One - IP address

prompt> **sntpclient set server ipaddress 129.6.15.28**

Example Two - hostname

prompt> **sntpclient set server hostname time-a.nist.gov**

1.282. sntpclient set timeout

Syntax

sntpclient set timeout <0-30>

Description

This command sets the received packet response timeout value (in seconds) upon sync request initiation. After timeout, if the *sntpclient retry* command value is set, an attempt will be retried.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
0-30	Sets the received packet response timeout value (in seconds). This can be any value between 0 and 30.	5 seconds

Example

prompt> **sntpclient set timeout 10**

1.283. sntpclient set timezone

Syntax

sntpclient set timezone <abbreviation>

Description

This command sets the local time zone abbreviation as a parameter and configures the local system to be up to + 13 hours of the Universal Time Coordinate (UTC). Sixty-four of the worlds most prominent time zones are represented (including those using standard time and summer/daylight savings time).

Options

The following table gives the 64 time zone abbreviations that you can use in this command. to set the timezone difference for the ISOS timer. The table also contains the difference in time (in hours and

minutes) from the UTC, and a description of the area of the world (from west to east) where the time difference is calculated from:

Example

In the example below, the time zone is set to Unites States Eastern Standard Time, which is five hours earlier than UTC (-0500):

prompt> **sntpclient set timezone EST**

1.284. sntpclient show association

Syntax

sntpclient show association

Description

This command lists the server configuration for the SNTP client with the timeserver address and displays whether or not the client is synchronized with the association server.

Examples

Example One - IP address

prompt> **sntpclient show association**
Time Reference Server IP address: 129.6.15.28
** Local clock synchronized with this server.

Example Two - hostname

prompt> **sntpclient show association**
Time Reference Server Hostname: time-a.nist.gov
** Local clock synchronized with this server.

1.285. sntp show status

Syntax

sntpclient show status

Description

This command displays the SNTP client status information.

Example

prompt> **sntpclient show status**
Clock Synchronized TRUE
SNTP Standard Version Number: 4
SNTP Mode(s) Configured: Unicast Broadcast
Local Time: Tuesday, 28 Aug, 2001 - 14:39:25
Local Timezone: EDT, Eastern Daylight Time
Time Difference +-VTC: -4:00
Precision: 1/16384 of a second
Root Dispersion: +0.2342 second(s)
Server Reference ID: GPS.
Round Trip Delay: 2 second(s)
Local Clock Offset: -1 second(s)
Resync Poll Interval 15 minute(s)
Packet Retry Timeout: 5 seconds
Packet Retry Attempts: 3

1.286. sntpclient sync

Syntax

sntpclient sync

Description

This command forces the SNTP client to immediately synchronize the local time with the server located in the association list (if unicast) or, if anycast is enabled, initiate an anycast sequence to the network.

Example

prompt> **sntpclient sync**

SyslogClient CLI Command

1.287. syslogClient set hostname

Syntax

syslogClient set hostname <hostName>

Description

Set the hostname for syslog client

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
hostName	Set the host name to use when sending out syslog messages.	N/A

Example

prompt> **syslogClient set hostname name**

1.288. syslogClient set receiver

Syntax

syslogClient set receiver <receiveripaddress>

Description

Set the target receiver for syslog receiver.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
receiver	Set the ip address to send syslog messages to.	N/A

Example

prompt> **syslogClient set receiver 192.168.2.46**

1.289. syslogClient set severity

Syntax

syslogClient set severity {alert|critical|emergencey|error|informational|notice|warning}

Description

Set the minium severity level above which syslog client can send message.

Example

prompt> **syslogClient set severity alert**

1.290. syslogClient show hostname

Syntax

syslogClient show hostname

Description

Show the hostname for syslog client.

Example

prompt> **syslogClient show hostname**
Host Name : name

1.291. syslogClient show receiver

Syntax

syslogClient show receiver

Description

Show the target receiver for syslog receiver.

Example

prompt> **syslogClient show receiver**
Receiver Ip Address: 192.168.2.46

1.292. syslogClient show severity

Syntax

syslogClient show severity

Description

Show the severity level set in syslog client.

Example

prompt> **syslogClient show severity**
Severity: informational

System CLI commands
This chapter describes the System CLI commands.

1.293. system add user

Syntax
system add user <name> [“comment”]

Description
This command adds a user to the system. Only superusers can use this command.

Default setting
The default setting in the table below are applied to new accounts added using the *system add user* command.

Option	Default setting
dialin to the system	enabled
login to the system	disabled
configuration permissions	disabled
access permissions	default user

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A unique user name made up of more than one character that identifies an individual user and lets the user access the system.	N/A
comment	An optional comment about the user that is displayed when you type the command *system list users* and *system list logins*.	No comment added

Example
prompt> **system add user <name> [“comment”]**

1.294. system config backup

Syntax
system config backup

Description
This command saves the system configuration to a file. After you enter the *sytem config backup* command, the configuration information is automatically saved in the *//isfs/im.conf.backup* and //flashfs/*im.conf.backup* files. To prevent a user from overwriting the system with their own configuration, only superusers can use this command.

Examples
prompt> **system config backup**
Saving configuration to files //isfs/im.conf.backup and //flashfs/im.conf.backup.

1.295. system config restore

Syntax
system config restore {backup |minimal}

Description

This command tries to restore all system modules; if you do not have all modules installed, the CLI will display a message telling you which modules could not be restored. The following options are available:

- superuser, engineer and default users can restore their backup configuration from the *//isfs/im.conf.backup* file.

- superuser can restore the factory defaults from *//isfs/im.conf.factory*.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
backup	Restores the backup configuration from the *//isfs/im.conf.backup* file.	N/A
factory	Restores the factory default configuration from the *//isfs/im.conf.factory* file. Only superusers can use this command.	N/A

Warning: Restore to factory default will change the IP address settings !

Example

prompt> **system config restore backup**
Restoring backup configuration //isfs/im.conf.backup

1.296. system config save

Syntax

system config save

Description

This command saves the system configuration in the *im.conf* file in FlashFS. This allows all users to create their own backup files. Superuser, engineer and default users can use this command.

Example

prompt> **system config save**
Wait for 'configurataion saved' message...
-->
Saving configuration...
Configuration saved.

1.297. system delete user

Syntax

system delete user <name>

Description

This command deletes a user that has been added to the system using the *system add user* command or the *system add login* command. Only superusers can use this command.

Options

The following table gives the range of values for each option which can be specified with this command

and a default value (if applicable).

Option	Description	Default value
name	The name of an existing user.	N/A

Example

prompt> **system delete user ckearns**

1.298. system info

Syntax

system info

Description

This command displays the vendor ID, URL, base MAC address and hardware and software version details that you are using.

Example

prompt> **system info**
Global System Configuration:
vendor:Tailyn
URL: http://www.tailyn.com.tw
MAC address: ##:##:##:##:##:##
Hardware version: BD62x1 BSP v1.0 / He100/2xx CSP v2.3
Software version: 1.0.0.2

1.299. system legal

Syntax

system legal

Description

This command displays copyright information about the software that you are using.

Example

prompt> **system legal**

1.300. system list errors

Syntax

system list errors

Description

This command displays a system error log. The error log contains the following information:

- the time (in minutes) that an error was made, calculated from the start of your session
- the module that was affected by the error
- a brief overview of the error itself

Example

prompt> **system list errors**
Error log:
When | Who | What
--- ----|--------------|----------
104 | webserver | webserver:Failed to create node type 'ImRfc1483'

104 | webserver | webserver:Invalid argument:Failed to open port a4 (may already be in use, or invalid port name)

1.301. system list openfiles

Syntax

system list openfiles <name>

Description

This command allows you to display low-level debug information about specific open file handles.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	The name of a file which has open file handles associated with it.	N/A

Example

prompt> **system list openfiles bun**
qid devuse appuse colour flags lasterrno
console 0000004b 00000000 00400000 3 0
console 00000027 00000000 00400000 5 0
console 00000003 00000000 00400000 5 0

1.302. system list users

Syntax

system list users

Description

This command displays a list of users and logins added to the system using the *system add user* command. The list contains the following information:

- user ID number
- user name
- configuration permissions (enabled or disabled)
- dialin permissions (enabled or disabled)
- access level (superuser, engineer, default)
- comment (any comments that were included when the user was added to the system)

Example

prompt> **system list users**
Users:
May May Access
ID	Name	Conf.	Dialin	Level	Comment
1	admin	ENABLED	disabled	superuser	Default admin user

1.303. system log

Syntax

system log {nothing|warnings|info|trace|entryexit|all}

Description

This command sets the level of output that is displayed by the CLI for various modules. Setting a level also implicitly displays the level(s) below it.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
nothing	No extra output is displayed.	N/A
warnings	Non-fatal errors are displayed.	N/A
info	Certain program messages are displayed. Also displays the values for the *warnings* option.	N/A
trace	Detailed trace output is displayed. Also displays the values for *info* and *warnings* options.	N/A
entryexit	A message is displayed every time a function call is entered or left. Also displays the values for *trace*, *info* and *warnings* options.	N/A
all	All output is displayed. Also displays the values for *entryexit, trace, info* and *warnings* options.	N/A

Example

prompt> **system log all**

1.304. system log enable|disable

Syntax

system log {enable|disable} <module> <category>

Description

This command enables/disables the tracing support output that is displayed by the CLI for a specific module and module category. The command is used for debugging purposes. The available values for module and category are displayed by the *system log list* command. The current list of supported modules is *RIP* and *IP*. The command only supports modules that are present in the current image that you are using. Each individual module has its own specific module category. The output produced when a particular option is enabled depends on that option, and on the trace statements in the module which are executed. The general purpose of this tracing is to:

- show how data packets pass through the system
- demonstrate how packets are processed and what they contain
- display any error conditions that occur

For example *ip rawip* tracing shows that an IP packet has been received, sent or discarded due to an error. Brief details of the packet are displayed to identify it. The RIP and IP modules provide separate categories which are enabled and disabled independently. For example, if you enable *ip rawip*, it does not affect *ip udp*, and so on.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enable	Enables tracing support output for a specified specific module and module category.	disable
disable	Disables tracing support output for a specified specific module and module category.	

Examples

prompt> **system log enable rip rx**
enabled logging for rip rx

1.305. system log list

Syntax

system log list [<module>]

Description

The *system log list* command displays the tracing options for the modules available in the current image that you are using. The *system log list module* command displays the tracing options for an individual module specified in the command. The command only displays modules and categories that are present in the current image that you are using.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
module	The name of a module that exists in your current image build. This can be either RIP, IP or IPoA, depending on the modules that you have present in your image build.	N/A

Examples

Example One - *system log list*

prompt> **system log list**
ip l2cyan (disabled)
rip errors (disabled)
rip rx (disabled)
rip tx (disabled)
ip socket (disabled)
ip tcp (disabled)
ip udp (disabled)
ip icmp (disabled)
ip rawip (disabled)
ip arp (disabled)

Example Two - *system log list <module>*

prompt> **system log list ip**
ip l2cyan (disabled)
ip socket (disabled)
ip tcp (disabled)
ip udp (disabled)
ip icmp (disabled)
ip rawip (disabled)
ip arp (disabled)

1.306. system restart

Syntax

system restart

Description

This command restarts your system. It has the same effect as pressing the reset button on your router card.

Example

prompt> **system restart**

1.307. system set user access

Syntax

system set user <name> access {superuser|engineer|default}

Description

This command sets the access permissions of a user who has been added to the system using the *system add user* command. Only superusers can use this command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	The name of an existing user.	N/A
superuser/ engineer/ default	Allows you to set the access permissions for a user.	default

Example

prompt> **system set user ckearns access default**

1.308. system set user mayconfigure

Syntax

system set user <name> mayconfigure {enabled|disabled}

Description

This command sets configuration permissions for a user who has been added to the system using the *add system user* command. Only superusers can use this command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	The name of an existing user.	N/A
enabled/ disabled	Determines whether or not a user can configure the system.	disabled

Example

prompt> **system set user ckearns mayconfigure enabled**

1.309. system set user maydialin

Syntax

system set user <name> maydialin {enabled|disabled}

Description

This command sets dial in permissions for a user who has been added to the system using the *system add user* command. Only superusers can use this command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	The name of an existing user.	N/A
enabled/ disabled	Determines whether or not a user can dialin to the system.	enabled

Example

prompt> **system set user ckearns maydialin enabled**

TCP/IP CLI commands
This chapter describes the TCP/IP CLI commands.

1.310. ip add interface

Syntax
ip add interface <name> [<ipaddress> [<netmask>]]

Description
This command adds a named interface and optionally sets its IP address. The IP address is not mandatory at this stage, but if it is not specified in this command, the interface will be unconfigured. There are three ways that the IP address can be set later:

- using the *ip set interface ipaddress* command
- you can set the interface to obtain its configuration via Dynamic Host Configuration Protocol (DHCP) using the *ip set interface dhcp enabled* command. By default, DHCP is disabled.
- this interface can obtain its IP configuration via PPP IPCP (Internet Protocol Control Protocol) negotiation. The IP stack automatically creates a loopback interface for address 127.0.0.1 subnet mask 255.0.0.0. This interface is not displayed by the *ip list interfaces* command. You can use this command to add unnumbered interfaces.

Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the ip interface. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit.	N/A
Ipaddress	The IP address of the interface displayed in the following format: 192.168.102.3 If the IP address is set to the special value *0.0.0.0*, the interface is marked as unconfigured. This value is used when the interface address is obtained automatically. For unnumbered interface, the IP address parameter is used to specify the router-id of the interface. The router-id should be the same as the IP address of one of the router's numbered interfaces.	0.0.0.0
netmask	The netmask address of the interface displayed in the following format: 255.255.255.0 The special value 255.255.255.255 is used to indicate an unnumbered interface. An unnumbered interface is configured by setting the IP address to the interface's router-id value, and setting netmask to 255.255.255.255.	If no IP address is supplied, the natural mask of the IP address is used.

Example
prompt> **ip add interface ip1 192.168.103.3 255.255.255.0**

1.311. ip add route

Syntax
ip add route <name> <dest_ip> <netmask> {[gateway<gateway_ip>]|[interface <interface>]}

Description

This command creates a static route to a destination network address via a gateway device or an existing interface. It also allows you to create a default route. You can only create one default route. A default route will **not** be created if you have already created a default route using the *ip add defaultroute gateway* command or the *ip add defaultroute interface* command. A route specifies a destination network (or single host), together with a mask to indicate what range of addresses the network covers, and a next-hop gateway address or interface. If there is a choice of routes for a destination, the route with the most specific mask is chosen. Routes are used when sending datagrams as well as forwarding them, so they are not relevant only to routers. However, a system with a single interface is likely to have a single route as a default route to the router on the network that it most often needs to use. If the interface can communicate more efficiently with a particular destination by using a different router, then it will learn this fact from an Internet Control Message Protocol (ICMP) redirect message.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
Name	route. It can be made up of one or more letters or a combination of letters and digits, but it cannot start with a digit. To create a default static route to a destination address, type *default* as the route name. You can only create one route called default.	N/A
dest_ip	The IP address of the destination network displayed in the following format: 192.168.102.3	N/A
Netmask	The destination netmask address displayed in the following format: 255.255.255.0	N/A
gateway_ip	The IP address of the gateway that this route will use, displayed in the following format: 192.168.102.3	N/A
interface	The name of the existing interface that this route will use. To display interface names, use the *ip list interfaces* command.	N/A

Examples

There are two examples in this section. Example 1 routes through a gateway. Example 2 is a default route.

Example 1

prompt> **ip add route route1 192.168.103.3 255.255.255.0 gateway 192.168.102.3**

Example 2

prompt> **ip add route default 0.0.0.0 0.0.0.0 interface ip1**

1.312. ip add defaultroute gateway

Syntax

ip add defaultroute gateway <gateway_ip>

Description

This command creates a default route. It acts as a shortcut command that you can use instead of typing the following:

ip add route default 0.0.0.0 0.0.0.0 gateway 192.168.103.3

You can only create one default route. A default route will **not** be created if you have already created a default route using the *ip add route* command or the *ip add defaultroute interface* command. If you want RIP to advertise a default route with a default cost metric, see the *ip set rip advertisedefault* and *ip set rip*

defaultroutecost commands.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
gateway_ip	The IP address of the gateway that this route will use by default, displayed in the following format: 192.168.102.3	N/A

Example

prompt> **ip add defaultroute gateway 192.168.103.3**

1.313. ip add defaultroute interface

Syntax

ip add defaultroute interface <interface>

Description

This command creates a default route. It acts as a shortcut command that you can use instead of typing the following:

ip add route default 0.0.0.0 0.0.0.0 interface ip3

You can only create one default route. A default route will **not** be created if you have already created a default route using the *ip add route* command or the *ip add defaultroute gateway* command. If you want RIP to advertise a default route with a default cost metric, see the *ip set rip advertisedefault* and *ip set rip defaultroutecost* commands.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
interface	The name of the existing interface that this route will use. To display interface names, use the *ip list interfaces* command.	N/A

Example

prompt> **ip add defaultroute interface ip3**

1.314. ip attach

Syntax

ip attach {<name>|<number>} <transport>

Description

This command attaches an existing transport to an existing IP interface (e.g., a bridge or router) so that data can be transported via the selected transport method. This command implicitly enables the transport being attached.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A

number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	N/A
transport	A name that identifies an existing transport.	N/A

Example

In the example below, *eth1* is the name of an ethernet transport created using the *ethernet add transport* command:

prompt> **ip attach ip1 eth1**

1.315. ip attachbridge

Syntax

ip attachbridge {<name>|<number>}

Description

This command attaches the bridge to the router via an existing IP interface.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	N/A

Example

prompt> **ip attachbridge 2**

1.316. ip clear interfaces

Syntax

ip clear interfaces

Description

This command clears all IP interfaces that were created using the *ip add interface* command.

Example

prompt> **ip clear interfaces**

1.317. ip clear riproutes

Syntax

ip clear riproutes

Description

This command deletes all the existing dynamic routes that have been obtained from RIP. It does not delete the static routes; see the *ip clear routes* command.

Example

prompt> **ip clear riproutes**

1.318. ip clear routes

Syntax

ip clear routes

Description

This command clears all static routes that were created using the *ip add route* command.

Example

prompt> **ip clear routes**

1.319. ip delete interface

Syntax

ip delete interface {<name>|<number>}

Description

This command deletes a single IP interface that was created using the *ip add interface* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command.The number appears in the first column under the heading *ID*.	N/A

Example

prompt> **ip delete interface ip1**

1.320. ip delete route

Syntax

ip delete route {<name>|<number>}

Description

This command deletes a single route that was created using the *ip add route* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing route. To display route names, use the *ip list routes* command.	N/A
number	A number that identifies an existing route. To display route numbers, use the *ip list routes* command. The number appears in the first column under the heading *ID*.	N/A

Example

prompt> **ip delete route route1**

1.321. ip detach

Syntax

ip detach {<name>|<number>}

Description

This command detaches a transport from an IP interface which was previously attached using the *ip attach interface* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	N/A

Example

prompt> **ip detach ip1**

1.322. ip interface add secondaryipaddress

Syntax

ip interface {<name>|<number>} add secondaryipaddress <ipaddress> [<netmask>]

Description

This command adds a secondary IP address to an existing IP interface. A secondary address may be used to create an extra IP address on an interface for management purposes, or to allow the IP stack to route between two subnets on the same interface. The functionality of secondary IP addresses depends on several parameters including the type of IP interface and the netmask: if a secondary address is on the **same** subnet as the primary interface address, you do not need to specify a subnet mask for that secondary address. This applies to all interface types.

- if a secondary address is on a **different** subnet to the primary address, and the interface is Ethernet or a transport using a bridged encapsulation, you must specify the subnet mask. The IP stack will listen on the new address for connections to local services (e.g., for management purposes), and will also route packets to the new subnet.
- if a secondary address is on a **different** subnet to the primary address, and the interface is a point-to-point interface, specifying a netmask is optional.
- for the same behavior as described for Ethernet interfaces above, the subnet mask **should** be specified.
- If the subnet mask is **not** specified, the IP address will not be associated with any subnet, but will still be recognized as one of the IP stack's own addresses for local traffic. The ability to specify a subnet mask with a secondary address is still supported, but in ISOS 8.2 (Service Release 2), this is **superseded** by the functionality of virtual interfaces. You should use virtual interfaces instead. Support for adding secondary IP addresses including subnet mask specification will be withdrawn in a future

ISOS release.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	N/A
ipaddress	A secondary IP address that you want to add to the main IP interface. You can add any number of secondary IP addresses. The IP address is displayed in the following format: 192.168.102.3 To display the secondary IP addresses, use the *ip interface list secondaryipaddresses* command.	N/A
netmask	The netmask of the secondary IP address displayed in the following format: 255.255.255.0 To display the secondary IP addresses, use the *ip interface list secondaryipaddresses* command.	none specified

Example

prompt> **ip interface ip1 add secondaryipaddress 192.168.102.3 255.255.255.0**

1.323. ip interface clear secondaryipaddresses

Syntax

ip interface {<name>|<number>} clear secondaryipaddresses

Description

This command deletes all additional IP addresses that have been added to an existing IP interface using the *ip interface add secondaryipaddress* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	N/A

Example

prompt> **ip interface ip1 clear secondaryipaddresses**

1.324. ip interface delete secondaryipaddress

Syntax

ip interface {<name>|<number>} delete secondaryipaddress <secondaryipaddress number>

Description

This command deletes a single secondary IP address that has previously been added to an existing IP interface using the *ip interface add secondaryipaddress* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	N/A
Secondary ipaddress number	The number that identifies a secondary IP address that you want to delete from the main IP interface. To display secondary IP address numbers, use the *ip interface list secondaryipaddresses* command. The number appears in the first column under the heading *ID*.	N/A

Example

prompt> **ip interface ip1 delete secondaryipaddress 1**

1.325. ip interface list secondaryipaddresses

Syntax

ip interface {<name>|<number>} list secondaryipaddresses

Description

This command lists the secondary IP addresses (and netmasks if applicable) that have been added to an existing IP interface using the *ip interface add secondaryipaddress* command.

29.32.3 Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	N/A

Example

In the example output below, secondary IP addresses without netmasks associated with them appear as 0.0.0.0 by default.

prompt> **ip interface ip1 list secondaryipaddresses**

ID | IP Address | Netmask

```
---|---------------------------------
1 | 192.168.104.6 | 255.255.255.0
2 | 192.168.103.4 | 0.0.0.0
3 | 192.168.103.2 | 0.0.0.0
-----------------------------------------
```

1.326. ip list arpentries

Syntax

ip list arpentries

Description

This command displays the ARP table which lists the following information:

- IP addresses and corresponding MAC addresses obtained by ARP.
- IP interface on which the host is connected
- Static status - ‘no’ for dynamically generated ARP entries; ‘yes’ for static entries added by the user.

Example

prompt> **ip list arpentries**

IP ARP table entries:

```
IP address    | MAC address          | Interface    | Static
--------------|----------------------|--------------|--------
10.10.10.10 | 00:20:2b:e0:03:87    | 3            | no
--------------|----------------------|--------------|--------
20.20.20.20 | 00:20:2b:03:0a:72    | 2            | no
--------------|----------------------|--------------|--------
30.30.30.30 | 00:20:2b:03:09:c4    | 1            | no
-----------------------------------------------------------------
```

1.327. ip list connections

Syntax

ip list connections

Description

This command lists the active TCP/UDP connections in use by applications running on the device. It displays the following information:

- Protocol type (TCP or UDP)
- Local connection address
- Remote connection address
- Connection state for TCP connections

This command does not show raw socket connections or UDP connections opened internally within the IP stack.

Example

The example below shows an active telnet connection, WebServer,
TFTP server and SNMP:

prompt> **ip list connections**

Local TCP/UDP connections:

```
Proto | Local address      | Remote address       | State
-------|--------------------|----------------------|--------------------
```

```
tcp   | 192.168.91.19:23 | 192.168.91.18:1080 | ESTABLISHED
tcp   | *:80             | *:*                | LISTEN
udp   | *:69             | *:*                |
udp   | *:161            | *:*                |
-------------------------------------------------------------------------------
```

1.328. ip list interfaces

Syntax

ip list interfaces

Description

This command lists information about IP interfaces that were added using the *ip add interface* command. The following information is displayed:

- interface ID numbers
- interface names
- IP addresses (if previously specified)
- DHCP status
- Whether a transport is attached to the interface, and if so, the name of the transport
- Whether a virtual interface is attached to a real interface. The name of the attached virtual interface is displayed in the *Transport* column in square brackets, for example [ip2]

Example

prompt> **ip list interfaces**

```
IP Interfaces:
ID | Name        | IP Address     | DHCP    | Transport
---|-------------|----------------|---------|--------------
1  | ppp_device | 192.168.102.2 | disabled | pppoe1
2  | ip2         | 192.168.102.3 | disabled | Not attached
3  | ip_real     | 192.168.101.2 | disabled | ethernet1
4  | ip_virtual  | 192.168.150.1 | disabled | [ip_real]
-------------------------------------------------------------------
```

1.329. ip list riproutes

Syntax

ip list riproutes

Description

This command lists information about the routes that have been obtained from RIP. It displays the following:

- destination IP addresses
- destination netmask address
- gateway address
- cost - The number of hops counted as the cost of the route.
- timeout - the number of seconds that this RIP route will remain in the routing table unless updated by RIP.
- source interface - the name of the existing interface that this route uses

Example

prompt> **ip list riproutes**

```
IP RIP routes:
Destination      | Mask            | Gateway      | Cost | Time | Source
-----------------|-----------------|--------------|------|------|----------
192.168.101.1 | 255.255.255.0 | 10.10.10.10 | 1      | 3000 | ip2
-------------------------------------------------------------------------------
```

1.330. ip list routes

Syntax

ip list routes

Description

This command lists information about existing routes. It displays the ID, name, destination IP address (if applicable), netmask address (if applicable) and gateway address or interface name (whichever is applicable).

- route ID numbers
- route names
- destination IP addresses (if previously specified)
- destination netmask address (if previously specified)
- Either the gateway address or the name of the destination interface (whichever is set)

Example

prompt> **ip list routes**

```
IP routes:
ID | Name    | Destination       | Netmask         | Gateway/Interface
---|---------|-------------------|-----------------|-----------------------
2    | route2    | 192.168.102.3 | 255.255.255.0 | ip1
1    | route1    | 192.168.50.50 | 255.255.255.0 | 192.168.68.68
-------------------------------------------------------------------------------
```

1.331. ip ping

Syntax

ip ping <dest-address>

Description

This command pings a specified destination IP address. You can only ping IP addresses. You can **not** ping host names using DNS client.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
dest-address	The address of the destination machine that you want to ping, displayed in the following format: 192.168.102.3	N/A

Example

prompt> **ip ping 192.168.102.3**

ip: ping - reply received from 192.168.102.3

If ping was unsuccessful, the following output is displayed:

ip: ping - no reply received.

1.332. ip set interface ipaddress

Syntax

ip set interface {<name>|<number>} ipaddress <ipaddress> [<netmask>]

Description

This command sets the IP address for an existing IP interface.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	N/A
Ip address	The IP address of the interface displayed in the following format: 192.168.102.3 If the IP address is set to the special value *0.0.0.0*, the interface is marked as unconfigured. This value is used when the interface address is obtained automatically. For unnumbered interfaces, the IP address parameter is used to specify the router-id of the interface. The router-id should be the same as the IP address of one of the router's numbered interfaces.	0.0.0.0
netmask	The netmask address of the interface displayed in the following format: 255.255.255.0 The special value 255.255.255.255 is used to indicate an unnumbered interface. An unnumbered interface is configured by setting the IP address to the interface's router-id value, and setting netmask to 255.255.255.255.	If no IP address is supplied, the natural mask of the IP address is used.

Example

prompt> **ip set interface ip4 ipaddress 192.168.102.3 255.255.255.0**

1.333. ip set interface netmask

Syntax

ip set interface {<name>|<number>} netmask <netmask>

Description

This command sets the netmask for an existing IP interface.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A

number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	N/A
netmask	The netmask address of the interface displayed in the following format: 255.255.255.0 The special value 255.255.255.255 is used to indicate an unnumbered interface. An unnumbered interface is configured by setting the IP address to the interface's router-id value, and setting netmask to 255.255.255.255.	N/A

Example

prompt> **ip set interface ip6 netmask 255.255.255.0**

1.334. ip set interface mtu

Syntax

ip set interface {<name>|<number>} mtu <mtu>

Description

This command sets the MTU (Maximum Transmission Unit) for an existing IP interface.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	1500
mtu	Maximum Transmission Unit: maximum packet size (in bytes) that an interface can handle. The MTU should be set to a value appropriate for the transport attached to the interface (typically from 576 to 1500 bytes). For example, Ethernet and most other transports support an MTU of 1500 bytes, whereas PPPoE supports an MTU of 1492 bytes.	1500

Example

prompt> **ip set interface ip2 mtu 800**

1.335. ip set interface dhcp

Syntax

ip set interface {<name>|<number>} dhcp {enabled|disabled}

Description

This command specifies whether a named interface should obtain its configuration via DHCP.

Options

The following table gives the range of values for each option which can be specified with this command

and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	N/A
enabled	The interface obtains its configuration information from DHCP client.	disabled
disabled	The interface does not use DHCP client configuration information.	

Example

prompt> **ip set interface ip2 dhcp enabled**

1.336. ip set interface rip accept

Syntax

ip set interface {<name>|<number>} rip accept
{none|v1|v2|all}

Description

This command specifies whether or not an existing interface accepts RIP messages. You can specify what version of RIP messages are accepted by the interface. When receiving RIP v1 messages, the IP stack tries to use the information it has available to determine the appropriate subnet mask for the addresses received.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interface* command. The number appears in the first column under the heading *ID*.	N/A
none	The interface does not accept RIP messages.	none
v1	The interface only accepts RIP version 1 messages (RFC1058).	
v2	The interface only accepts RIP version 2 messages (RFC1723).	
all	The interface accepts RIP version 1 (RFC1058) and RIP version 2 (RFC1723) messages.	

Example

prompt> **ip set interface ip3 rip accept none**

1.337. ip set interface rip multicast

Syntax

ip set interface {<name>|<number>} rip multicast {enabled | disabled}

Description

This command allows you to enable/disable whether RIP version 2 messages are sent via multicast.
RIP version 2 messages sent via multicast are only received by the hosts on the network that have a multicast network address. If this command is disabled, RIP version 2 messages are sent via broadcast and are received by all the hosts on the network. You need to set RIP to send v2 messages using the *ip set interface rip send* command in order for the *ip set interface rip multicast enabled* command to send version 2 messages via multicast.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	N/A
enabled	Allows RIP version 2 messages to be sent via multicast.	disabled
disabled	Disables RIP version 2 messages being sent via multicast. Messages are sent via broadcast instead.	

Example

prompt> **ip set interface ip1 rip multicast enabled**

1.338. ip set interface rip send

Syntax

ip set interface {<name>|<number>} rip send {none|v1|v2|all}

Description

This command specifies whether or not an existing interface can send RIP messages. You can specify which version of RIP messages will broadcast routing information on the interface. Routing information is broadcast every 30 seconds or when the RIP routing table is changed. RIP version 1 does not allow specification of subnet masks; a RIP version 1 route that appears to be to an individual host might in fact be to a subnet, and treating it as a route to the whole network may be the best way to make use of the information.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	N/A

rip send none	The interface does not accept RIP messages.	rip send none (this command affects all interfaces except loopback interfaces)
rip send v1	The interface only sends RIP version 1 messages (RFC1058)	
rip send v2	The interface only sends RIP version 2 messages (RFC1723). If set, RIP version 2 is used on all non-loopback interfaces.	
rip send all	The interface sends RIP version 1 (RFC1058) and RIP version 2 (RFC1723) messages.	

Example

prompt> **ip set interface ip1 rip send v1**

1.339. ip set interface rip password

Syntax

ip set interface <name> rip password <password>

Description

This command specifies the password can send RIP messages.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
password	RIP password	N/A

Example

prompt> **ip set interface ip1 rip password xxxx**

1.340. ip set interface rip Auth

Syntax

ip set interface <name> rip Auth {enable| disable}

Description

This command allows you to enable/disable the RIP authentication.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A

Example

prompt> **ip set interface ip1 rip auth enable**

1.341. ip set interface tcpmssclamp

Syntax

ip set interface <name> tcpmssclamp {enabled|disabled}

Description

This command enables/disables TCP MSS (Maximum Segment Size) Clamp functionality on an existing IP interface. When TCP MSS Clamp is enabled on an interface, all TCP traffic routed through that interface will be examined. If a TCP SYN (synchronize/start) segment is sent with a maximum segment size larger than the interface MTU (Maximum Transmission Unit), the MSS option will be rewritten in order to allow TCP traffic to pass through the interface without requiring fragmentation.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
enabled	TCP SYN segments routed through this interface will be examined and, if necessary, modified.	disabled
disabled	The IP stack will not examine or modify TCP traffic routed through this interface.	

Example

prompt> ip set interface ip2 tcpmssclamp enabled

1.342. ip set rip hostroutes

Syntax

ip set rip hostroutes {enabled | disabled}

Description

Specifies whether IP interfaces will accept RIP routes to specific routes. RIP version 1 does not allow specification of subnet masks; a RIP version 1 route that appears to be to an individual host might in fact be to a subnet, and treating it as a route to the whole network may be the best way to make use of the information. To display the current state of *rip hostroutes*, use the *ip show* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
Rip hostroutes enabled	Sets the *hostroutes* flag to *on*. The interface accepts RIP routes to specific routes.	Rip hostroutes disabled
rip hostroutes disabled	Sets the *hostroutes* flag to *off*: RIP version 1 routes to individual hosts are treated as routes to the network containing the host. RIP version 2 routes to individual hosts are ignored.	

Example

prompt> **ip set rip hostroutes enabled**

1.343. ip set rip poison

Syntax

ip set rip poison {enabled | disabled}

Description

Enables or disables the *poisoned reverse* flag. If this flag is on, ATMOS TCP/IP performs *poisoned reverse* as defined in RFC 1058; see that RFC for discussion. To display the current state of the *poisoned reverse* flag, use the *ip show* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
rip poison enabled	Sets the *poisoned reverse* flag to *on*. ATMOS TCP/IP performs poisoned reverse as defined in RFC 1058.	rip poison disabled
rip poison disabled	Sets the *poisoned reverse* flag to *off*:	

Example

prompt> **ip set rip poison enabled**

1.344. ip set route destination

Syntax

ip set route {<name>|<number>} destination <dest-network> <netmask>

Description

This command sets the destination network address of a route previously created using the *ip add route* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing route. To display route names, use the *ip list routes* command.	N/A
number	A number that identifies an existing route. To display route numbers, use the *ip list routes* command. The number appears in the first column under the heading *ID*.	N/A
dest-network	The IP address of the destination network displayed in the following format: 192.168.102.3	N/A
netmask	The destination netmask address displayed in the following format: 255.255.255.0	N/A

Example

prompt> **ip set route route1 destination 192.168.103.3 255.255.255.0**

1.345. ip set route gateway

Syntax

ip set route {<name>|<number>} gateway <gateway>

Description

This command sets the gateway address of a route previously created using the *ip add route* command. If you want the route to go directly to its destination and not via a gateway, specify *0.0.0.0* as the gateway.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing route. To display route names, use the *ip list routes* command.	N/A
number	A number that identifies an existing route. To display route numbers, use the *ip list routes* command. The numbers appear in the first column under the heading *ID*.	N/A
gateway	The IP address of the gateway that the IP routes through, displayed in the following format: 192.168.102.3 If you added a route directly to an interface, the gateway address is set by default to 0.0.0.0 so that no gateway is specified.	N/A

Example

prompt> **ip set route route1 gateway 192.168.102.3**

1.346. ip set route cost

Syntax

ip set route {<name>|<number>} cost <cost>

Description

This command sets the number of hops counted as the cost of the route for a route previously created using the *ip add route* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing route. To display route names, use the *ip list routes* command.	N/A
number	A number that identifies an existing route. To display route numbers, use the *ip list routes* command. The number appears in the first column under the heading *ID*.	N/A
cost	The number of hops counted as the cost of the route. This may affect the choice of route when the route is competing with routes acquired from RIP. (Using a mixture of RIP and static routing is not advised). The cost value can be any positive integer.	1

Example

prompt> **ip set route route1 cost 3**

1.347. ip set route interface

Syntax

ip set route {<name>|<number>} interface {<interface>|none}

Description

This command sets the interface used by a route previously created by the *ip add route* command. If you want the existing route to route to an address via a gateway device, use *none* so that no interface is set.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing route. To display route names, use the *ip list routes* command.	N/A
number	A number that identifies an existing route. To display route numbers, use the *ip list routes* command. The number appears in the first column under the heading *ID*.	N/A
interface	The name of the existing interface that the ip routes through, displayed in the following format: 192.168.102.3 To display interface names, use the *ip list interfaces* command.	N/A
none	No interface is set. This is used for routes that route via a gateway device instead of an interface.	N/A

Example

prompt> **ip set route r1 interface eth1**

1.348. ip show

Syntax

ip show

Description

Shows current RIP configuration and any other information global to the router.

Example

prompt> **ip show**
Global IP configuration:
Host routes: true
Poison reverse: false
Authentication: true
Auth password: vancouver

1.349. ip show interface

Syntax

ip show interface {<name>|<number>}

Description

This command displays the following information about a named interface:

- IP address and netmask address (if set). For virtual interfaces, the name of the real interface that the virtual interface is attached to is also displayed.
- MTU (Maximum Transmission Unit)
- Status of DHCP
- Status of TCP MSS Clamp
- Status of RIP send and RIP accept
- Status of RIP multicast

Options

The following table gives the range of values for each option which can be specified with this command

and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	N/A
number	A number that identifies an existing IP interface. To display interface numbers, use the *ip list interfaces* command. The number appears in the first column under the heading *ID*.	N/A

Examples

Real IP interface

prompt> **ip show interface ip2**
IP Interface: ip2
IP address: 192.168.102.3
Netmask: 255.255.255.0
MTU: 1500
DHCP: disabled
TCP MSS Clamp: disabled
Accept RIP V1: true
Send RIP V1: false
Accept RIP V2: true
Send RIP V2: false
Multicast RIP V2: disabled

Virtual IP interface

prompt> **ip show interface ip3**
IP Interface: ip3 - virtual [ip2]
IP address: 192.168.50.10
Netmask: 255.255.255.0
MTU: 1500
DHCP: disabled
TCP MSS Clamp: disabled
Accept RIP V1: true
Send RIP V1: false
Accept RIP V2: true
Send RIP V2: false
Multicast RIP V2: disabled

1.350. ip show route

Syntax

ip show route {<name>|<number>}

Description

This command displays the following information about a named route:
• Destination IP address
• Netmask address
• Gateway IP address
• Cost: the number of hops counted as the cost of the route
• Interface name

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existingroute. To display route names, use the *ip list routes* command.	N/A
number	A number that identifies an existing route. To display route numbers, use the *ip list routes* command. The number appears in the first column under the heading *ID*.	N/A

Example

prompt> **ip show route route3**
IP route: route3
Destination: 192.168.102.3
Netmask: 255.255.255.0
Gateway: 192.168.108.3
Cost: 1
Interface:

1.351. ip show debuginfo

Syntax

ip show debuginfo

Description

This command displays the debug information about IP stack such as :
• Interface index
• Interface Name
• IP address and subnet mask
• Routing table
• etc…

Example

prompt> **ip show debuginfo**
Found IP stack.

Interfaces:

IfIndex: 0 Name: eth0 Addr: 192.168.7.15 Mask: 255.255.0.0
All addresses:
192.168.7.15 255.255.0.0
239.255.255.250 255.255.255.255
192.168.255.255 255.255.255.255
255.255.255.255 255.255.255.255
IGMP membership:
239.255.255.250
DHCP: disabled MSS Clamp: disabled
Rx Filter: present Tx Filter: present
IfType: ETHER
Device: //bun/port=ethernet

```
IfIndex:  1    Name: ppp-0          Addr: 0.0.0.0          Mask: 255.255.255.0
      All addresses:
           0.0.0.0          255.255.255.0
           255.255.255.255  255.255.255.255
      IGMP membership:
      DHCP: disabled           MSS Clamp: disabled
      Rx Filter: present       Tx Filter: present
      IfType: PTP
      Device: //ppp/DEVICE=1

IfIndex: 15    Name: loopback       Addr: 127.0.0.1        Mask: 255.0.0.0
      All addresses:
           127.0.0.1        255.0.0.0
           127.255.255.255  255.255.255.255
           255.255.255.255  255.255.255.255
      IGMP membership:
      DHCP: disabled           MSS Clamp: disabled
      Rx Filter: none          Tx Filter: none
      IfType: LOOP
      Device: (null)


Routing table:
-----------------
Dst: 127.255.255.255 / 32    Gw:    0.   0.   0.   0    If: 15   Cost:   1   Int: yes
Dst: 127.    0.   0.   1 / 32    Gw:    0.   0.   0.   0    If: 15   Cost:   1   Int: yes
Dst: 192.168.255.255 / 32    Gw:    0.   0.   0.   0    If:  0   Cost:   1   Int: yes
Dst: 192.168.    7. 15 / 32    Gw:    0.   0.   0.   0    If:  0   Cost:   1   Int: yes
Dst: 192.168.    0.   0 / 16    Gw:    0.   0.   0.   0    If:  0   Cost:   1   Int: no
Dst: 127.    0.   0.   0 / 8    Gw:    0.   0.   0.   0    If: 15   Cost:   1   Int: no

IGMP Proxy multicast forwarder:
----------------------------------------
      Upstream interface: none
      Group address       Interfaces


Compile time configuration:
-------------------------------------
      QOS support: disabled
      Checksum forwarded packets: no
      Multihomed routing: enabled
      Preference: routed traffic
      ATIC Layer 2: not present
```

TFTPC CLI Commands

This chapter describes CLI support for TFTP Client.

1.352. tftpc connect

Syntax

tftpc connect <host>

Description

This command allows you to specify the remote node name or IP address for the *host* node that will be used in subsequent client mode transfers.

This command is required before a client mode user first attempts to *put* or *get* a file, but need not be issued again unless you want to change the remote host node name or address.

Example

prompt> **tftpc connect 192.168.200.10**

1.353. tftpc disconnect

Syntax

tftpc disconnect

Description

This command disconnects the existing host that was specified in *tftpc connect* command.

Example

prompt> **tftpc disconnect**

1.354. tftpc get

Syntax

tftpc get <src> <dst>

Description

This command configures TFTP to retrieve a file from the remote host previously specified using *tftpc connect* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value for each option (if applicable).

Option	Description	Default value
src	The name of the file to be retrieved from the connected host.	N/A
dst	The name given to the file once it has been retrieved from source.	N/A

Example

prompt> **tftpc get im.conf //flashfs/im.conf**

1.355. tftpc put

Syntax

tftpc put <src> <dst>

Description

This command configures TFTP to transmit a file to the remote host previously specified using *tftpc connect* command.

Options

The following table gives the range of values for each option which can be specified with this command and a default value for each option (if applicable).

Option	Description	Default value
src	The name of the file to be transmitted from source.	N/A
dst	The name given to the file once it has been transmitted to the connected host.	N/A

Example

prompt> **tftpc put addresslistA ipaddresses**

Transports CLI commands
This chapter describes the Transports commands provided by the CLI:

1.356. transports clear

Syntax
transports clear
Description
This command deletes all transports that were created using the <*transport_module*> *add transport* command.
Example
prompt> **transports clear**

1.357. transports delete

Syntax
transports delete {<name>|<number>}
Description
This command deletes a single transport that was created using the <*transport_module*> *add transport* command.
Options
The following table gives the range of values for each option which can be specified with this command and a default value for each option (if applicable).

Option	Description	Default value
name	A name that identifies an existing transport. To display transport names, use the *transports list* command.	N/A
number	A number that identifies an existing transport. To display transport numbers, use the *transports list* command.	N/A

Example
prompt> **transports delete eth1**

1.358. transports list

Syntax
transports list
Description
This command lists all transports created during a session. It displays the following information about the transports:
- transport identification number
- transport name
- transport type (RFC1483, PPP, Ethernet, Frame Relay or IPoA)
- Number of transmitted/received packets for each transport
- VPI/VCI setting (RFC1483, PPP and IPoA transports only)

Example
prompt> **transports list**
Services:

```
ID | Name    | Type
----|-----------|---------------------------------------------------------------
1    | rfc1483 | RFC1483 | TxPkts:0/0 RxPkts:0/0 VPI/VCI:0/100
2    | pppoh2 | PPP          | TxPkts:0/0 RxPkts:0/0 VPI/VCI:0/101
3    | pppoh1 | PPP          | TxPkts:0/0 RxPkts:0/0 VPI/VCI:0/102
4    | pppoa2 | PPP          | TxPkts:0/0 RxPkts:0/0 VPI/VCI:0/103
5    | eth0      | Ethernet    | TxPkts:0/0 RxPkts:0/0
-------------------------------------------------------------------------------
```

1.359. transports show

Syntax

transports show {<name>|<number>}

Description

This command displays detailed information about an existing transport. The information displayed depends on the transport type selected. See below for examples of PPP and RFC1483 transport information.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing transport. To display transport names, use the *transports list* command.	N/A
number	A number that identifies an existing transport. To display transport numbers, use the *transports list* command.	N/A

Examples (PPPoH transport)

```
--> transports show ppp-0
PPP Status

Service
Creator             : WebAdmin
Description         : PPPoH Routed

PPP
Summary             : enabled, down
Server              : false
Create Route        : true
Specific Route      : false
Subnet Mask         : 0.0.0.0
Route Mask          : 0.0.0.0
Hdlc                : false
LLC                 : false
Lcp Max Configure   : 10
Lcp Max Failure     : 5
Lcp Max Terminate   : 2
```

```
Dialin Auth         : none
Dialout Username    :
Dialout Password    :

Confirmation Password :
Dialout Auth        : none
Dialout Username    :
Dialout Password    :

Confirmation Password :
Dialout Auth        : none
Interface ID        : 1
Magic Number        : 0
MRU                : 0
Ip Addr From IPCP    : true
Discover Primary DNS : true
Discover Secondary DNS : true
Give DNSto Relay    : true
Give DNSto Client   : true
Lcp Echo Every      : 10
Auto Connect        : false
Idle Timeout        : 0
Connect State       : connecting
Uptime             : 0
Idletime           : 0
If In Octets        : 0
If Out Octets       : 3480
If In Errors        : 0
If Out Errors       : 547
Packets Sent        : 580
Good Packets Received : 0
Enabled            : true

Termination         : Ip Interface: ppp-0


Hdlc Channel
Port               : hdlc


-->
```

User CLI commands
This chapter describes the User CLI commands.

1.360. user logout

Syntax
user logout
Description
This command logs you out of the system. Superuser, engineer and default users can use this command.
Example
prompt> **user logout**
Logging out.

1.361. user password

Syntax
user password
Description
This command allows you to change your user password. Superuser, engineer and default users can use this command.
Example
prompt> **user password**
Enter new password *****
Again to verify *****

1.362. user change

Syntax
user change <name>
Description
This command allows you to change your login to that of another named user. Superusers can use this command. When you change your login to that of a user with User or Guest access permissions, you lose your superuser privileges and inherit the access permissions of either the User or Guest user.
Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
name	A unique login name made up of more than one character that identifies an individual user and lets the user access the system.	N/A

Example
prompt> **user change admin**
You are now logged in as user ‘admin’ ...

Web Server CLI commands
This chapter describes the Web Server CLI commands.

1.363. webserver clear stats

Syntax
webserver clear stats
Description
This command sets all of the Web Server process counters to 0.
Example
prompt> **webserver clear stats**

See also 1.370 *webserver show info*.

1.364. webserver enable|disable

Syntax
webserver {enable|disable}
Description
This command enables or disables the Web Server process. By default, the Web Server process is enabled.
Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
enable	Enables the Web Server process.	enable
disable	Disables the Web Server process.	

Example
prompt> **webserver disable**
WebServer is disabled

1.365. webserver set interface

Syntax
webserver set interface <interface>
Description
This command specifies the name of an IP interface that an ISOS IGD (Internet Gateway Device) will use for UPnP (Universal Plug and Play) communication with other devices on the local area network. By default, your system creates an IP interface with an Ethernet transport attached to it. This interface is called *iplan,* and it is the default interface that UPnP uses for its communication. Once you have set the UPnP interface, the IGD monitors the interface. The IGD can handle changes to the interface definition (for example, if the IP address changes through a DCHP update, the IGD will use the newly assigned address). **You must save your configuration (see *system config save*) and restart your system (see *system restart*) to activate the Web Server settings.**
Options
The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
interface	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	iplan

Example

prompt> webserver set interface ip

See also 1.369 *webserver set upnpport.*

1.366. webserver set managementip

Syntax

webserver set managementip {ip-address}

Description

This command allows connection requests to be restricted to a set of IP addresses according to managementipmask (e.g. managementip = 172.16.10.0, managementipmask = 255.255.255.0, then from 172.16.10.1 to 172.16.10.254 are included in the management set of IP addresses), or only one IP address (e.g. managementip = 172.16.10.2, managementipmask = 255.255.255.255) or from any IP address (by setting the managementip and managemenentipmask to 0.0.0.0).

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
ip-address	The IP address that the Web Server will allow connection requests from. The IP address is displayed in the following format: 172.16.10.0	0.0.0.0

Example

prompt> **webserver set managementip 172.16.10.0**
Management IP address's subnet is (172.16.10.0& managementipmask)

See also 1.367 *webserver set managementipmask*

1.367. webserver set managementipmask

Syntax

webserver set managemenentipmask {netmask}

Description

This command allows connection requests to be restricted to a set of IP addresses according to managementip (e.g. managementip = 172.16.10.0, managementipmask = 255.255.255.0, then from 172.16.10.1 to 172.16.10.254 are included in the management set of IP addresses), or only one IP address (e.g. managementip = 172.16.10.2, managementipmask = 255.255.255.255) or from any IP address (by setting the managementip and managemenentipmask to 0.0.0.0).

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable)

Option	Description	Default value

netmask	The netmask address of the managementipmask displayed in the following format: 255.255.255.0.	0.0.0.0

Example

prompt> **webserver set managementipmask 255.255.255.0**
Management IP address's subnet is (managementip & 255.255.255.0)

See also 1.366 *webserver set managementip*

1.368. webserver set port

Syntax

webserver set port <port>

Description

This command sets the HTTP port number that the Web Server process will use to transfer data.

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
port	A valid port number that must be between 0 and 65535.	80

Example

prompt> **webserver set port 100**
HTTP port number is 100

See also 1.369 *webserver set upnpport.*

1.369. webserver set upnpport

Syntax

webserver set upnpport <port>

Description

This command sets the UPnP (Universal Plug and Play) port number that the Web Server process will use for UPnP communication. **You must save your configuration (see *system config save*) and restart your system (see *system restart*) to activate the Web Server settings.**

Options

The following table gives the range of values for each option which can be specified with this command and a default value (if applicable).

Option	Description	Default value
port	A valid UPnP port number that must be between 0 and 65535.	N/A

Example

prompt> **webserver set upnpport 280**

See also 1.368 *webserver set port.*

1.370. webserver show info

Syntax

webserver show info

Description

This command displays the following information about the Web Server process:

- EmWeb (Embedded Web Server) release details
- Web Server enabled status (true or false)
- Interface set
- HTTP port set
- UPnP port set
- Management IP address

Example

prompt> **webserver show info**
Web server configuration:
EmWeb release: R6_1_0
Enabled: true
Interface: iplan
HTTP port: 80
UPnP port: 280
Management IP address: 1.2.3.4

See also 1.363 *webserver clear stats.*

1.371. webserver show stats

Syntax

webserver show stats

Description

This command tells you how many bytes have been transmitted and received by the Web Server.

Example

prompt> **webserver show stats**
Web Server statistics:
Bytes transmitted: 2122
Bytes received: 0

See also 1.370 *webserver show info.*

Other commands

1.372. help

Syntax

help

Description

This command shows some help information.

Example

prompt> **help**

1.373. source

Syntax

source <filename>

Description

This command allows you to run a list of predefined commands stored in an existing file.

Example

prompt> **source <filename>**

```
Sourcing file '//isfs/myconfiguration.txt'...
--> ethernet clear transports
--> ethernet add transport eth1 ethernet
--> bridge add interface bridge1
--> bridge attach bridge1 eth1
--> framerelay add transport fr1 fr 171
--> framerelay set transport fr1 encapsulation bridgedether
--> bridge add interface bridge2
--> bridge attach bridge2 fr1
--> ethernet list transports

Ethernet transports:

 ID   |    Name    |    Port
------|------------|------------
   1  | eth1       | ethernet
------------------------------

--> bridge list interfaces

Bridge Interfaces:

 ID  |   Name     | Filter Type |    Transport
-----|----------|-------------|------------------
   1 | bridge2   | All         | fr1
   2 | bridge1   | All         | eth1
-------------------------------------------------

--> framerelay list transports

Frame Relay Transports:

 ID  |   Name   |   Port   |  DLCI  | Encapsulation
```

```
-----|---------|-------- |-------|------------------------
  1  | fr1     |   fr    |  171  | BridgedEther
-------------------------------------------------------
```

Appendix A: TFTP Console commands

This chapter describes the TFTP console commands.

A1. connect

Access permission

Users with ‘superuser’ access permission can use the command.

Syntax

connect <node name> | | <ipaddress>

Scope

Client mode only.

Description

The *connect* command is used to specify the remote node name or IP address for the *host* node that will be used in subsequent client mode transfers.
Either a *node name* may be entered, searched for in the *ipaddresses* configuration file, or an IP address in the form *abc.def.ghi.jkl*. If the *node name* is not recognised or the IP address does not convert correctly, an error is signalled.
The non-appearance of an error message after the command *does not* signify that the host node is accessible, only that the syntax of the command was appropriate.
This command is required before a client mode user first attempts to *put* or *get* a file, but need not be issued again unless you want to change the remote host node name or address.

Example

prompt> **connect 192.168.200.10**

A2. get

Access permission

Users with ‘superuser’ access permission can use the command.

Syntax

get <remote_file> [local_file]

Scope

Client mode only.

Description

The *get* command requests TFTP to retrieve a file from the remote host previously specified using the *connect* command.
Only files that fit within the file storage area within the session data can be retrieved. This means that it is not possible to initiate a software update from the client.
By default the file is named locally as the remote filename but by specifying a second filename an implicit rename is performed.

Example

prompt> **get ipaddresses**

A3. put

Access permission

Users with ‘superuser’ access permission can use the command.

Syntax

put [local_file] <remote_file>

Scope

Client mode only.

Description

The *put* command requests TFTP to transmit a file to the remote host previously specified using the *connect* command.

By default, the file is named remotely as the local filename but by specifying a second filename, an implicit rename is performed.

Example

prompt> **put foo.txt**