ISOS (8.2) CLI Reference Manual

DO-009430-PS (Issue 1, 08 Apr 2002)

Copyright

ISOS (8.2) CLI Reference Manual:

DO-009430-PS

GlobespanVirata Company Proprietary.

Trademarks

Contacting GlobespanVirata

For more information on GlobespanVirata, contact the offices below or visit our web site:

http://www.globespanvirata.com

GlobespanVirata Headquarters

100 Schulz Drive
Red Bank, NJ 07701
USA
phone: +1-888-855-4562

GlobespanVirata UK

Unit 230 Cambridge Science Park
Milton Road
Cambridge CB4 0WB
United Kingdom
phone: +44 1223 707400

Contents

14. Firewall CLI commands 173

21. PPPoA CLI commands 311

22. PPPoE CLI commands 383

24. PPTP CLI commands 501

25. RFC1483 CLI commands 517

26. Security CLI commands 541

27. SNTP CLI commands 571

28. System CLI commands 587

29. TCP/IP CLI commands 619

Appendix A: ATMOS Console Commands 705

Appendix B: Bridge Console Commands 729

Appendix C: BUN Console Commands 747

Appendix D: Chips Console Commands 771

Appendix E: DHCP-client Console Commands 783

Appendix J: TCP/IP Console commands 829

Appendix K: PPP Console Commands 845

1. About this Guide

This chapter tells you about:

- *The scope of this guide and its intended audience*
- *The typographical conventions used in this guide*
- *How to read and provide feedback about this guide*

The information contained in this guide must be read and fully understood before you attempt to use the product.

1.1 Introduction

This document is a reference guide for developers working with **Release 8.2** of ISOS. It describes the command line interface (CLI) for the ISOS software modules supported by the VMI. It also contains console commands for the most commonly used ISOS modules which are not yet fully supported by the VMI.

1.2 Scope

Developers should refer to the individual process *Functional Specifications* for more detailed and extensive information about each individual ISOS process.

This document does not provide extensive examples of how to configure the software. See the configuration chapters of the *User's Guide* supplied with your GlobespanVirata system for some detailed examples:

- *ISOS (8.2) User Guide: DO-009467-PS.*
- *ISOS (8.2) Switch Router User Guide: DO-009500-PS*

These guides also provide an introduction to using the CLI and describe the terminology and conventions it uses.

1.3 Audience

This document is targeted at Licensees evaluating or developing systems for ISOS release 8.2.

1.4 References

For more information on the ISOS software modules referred to in this manual, refer to the functional specification for each module. The functional specifications listed below are available from the Licensee Server

- *ISOS CLI Functional Specification: DO-008362-PS*
- *IP Stack Programmer's Guide: DO-010017-TC*
- *GlobespanVirata Transparent Bridge Functional Specification: DO-007087-PS*
- *BUN Devices: Ethernet Functional Specification: DO-007444-PS*
- *BUN Standard Attributes Functional Specification: DO-007441-PS*

- *BUN Devices: RFC1483 Functional Specification: DO-007605-PS*
- *ATMOS PPP Functional Specification: DO-007078-PS*
- *BUN Devices: Point to Point Protocol over Ethernet (PPPoE) Functional Specification: DO-008195-PS*
- *BUN Devices: Frame Relay Functional Specification: DO-008218-PS*
- *ATMOS PPTP Functional Specification: DO-007352-PS*
- *GlobespanVirata TFTP Functional Specification: DO-007137-PS*
- *ATMOS DHCP Server Functional Specification: DO-007343-PS*
- *ATMOS DHCP Client Functional Specification: DO-007309-PS*
- *ATMOS DHCP Relay Functional Specification: DO-007608-PS*
- *GlobespanVirata DNS Relay Functional Specification: DO-007692-PS*
- *GlobespanVirata DNS Client Functional Specification: DO-008322-PS*
- *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*
- *ATMOS OpenDSL Functional Specification: DO-008383-PS*
- *TR037 AutoPVC Functional Specification: DO-008520-PS*

1.5 Typographical conventions

Throughout this guide, the following typographical conventions are used to denote important information.

1.5.1 Text conventions

The following text conventions are used:

- *Text like this* is used to introduce a new term, to indicate menu options or to denote field and button names in GUI windows and dialogue boxes.
- **Text like this** is used to emphasize important points. For example: 'To keep your changes, you **must** save your work before quitting.'
- **`Text like this`** is used for text that you type as a command or entry to a field in a dialogue box. Variables to a command are shown in *`text like this`*.

- `Text like this` is used for text that you see on the screen in a terminal window. Variables to displayed text are shown in *`text like this`*.
- `<Text like this>` in angle brackets is used for denoting command line options. It indicates a mandatory argument.
- `[Text like this]` in square brackets is used for denoting command line options. It indicates an optional argument.
- Text in square brackets is used to indicate keyboard keys. For Example:

  'To reboot your computer, press [Ctrl]+[Alt]+[Del].'
- Type versus Enter; **Type** means type the text as shown in the instruction. **Enter** means type the text as indicated and then press [Enter].

1.5.2 Notes, Warnings and Cautions

The following symbols are used:

Warning - Indicates a hazard which may endanger equipment or personnel if the safety instruction is not observed.

Caution - Indicates a hazard which may cause damage to equipment if the safety instruction is not observed.

Note - Indicates general additional information about the operation of the equipment including safety information.

1.6 Reading this guide

1.6.1 Providing feedback on this guide

Please report problems and questions via the *Problem Reports* area on the GlobespanVirata licensee web server at:

When submitting feedback, please give the full title, part number and version number of the guide.

1.6.2 Reading this guide

This guide is available in two formats:

- PDF version - *ISOS (8.2) CLI Reference Manual: DO-009430-PS*

- HTML version - *ISOS (8.2) CLI Reference Manual (HTML format): DO-009514-LS*

If you are reading the PDF version of this guide, it can be printed for easy reference. However, it has been prepared for viewing on-line through a web browser (such as Internet Explorer™ or NetScape Navigator™), or Adobe Acrobat™.

Links to other sections of the PDF version or to other guides are marked in blue (although the links are black when printed on a standard laser printer.) Click on the link to view the associated section or document.

2. Introduction to the CLI

This chapter provides some basic instructions on how to use the CLI, and gives details of how the information in this manual is organized.

There is also an HTML version of this manual available from the GlobespanVirata Licensee Server. See the ISOS (8.2) CLI Reference Manual (HTML format): DO-009514-LS.

2.1 About the CLI

The CLI is the *Command Line Interface* used in ISOS releases later than 8.0. It largely replaces the *console commands* that were used in releases prior to the 8.0 release.

Some console commands are available for use if you have appropriate access permissions set. For details of access permissions, see *Access permissions to the CLI* on page 9.

The CLI provides a command line interface to the ISOS modules that are modelled in the VMI Information Model. For more details on using the command line interface, see *Using CLI and Console Commands* on page 8.

2.2 Using CLI and Console Commands

There are two types of command available for use in ISOS:

- *CLI commands* - the CLI command *replaces* the majority of console commands. For example, the console command:
  ```
  bridge device add
  ```
  is now the CLI command:
  ```
  bridge add interface
  ```
- *Console commands* - the console command *has not* been replaced by a CLI command. Users with appropriate access permissions (see *Access permissions to the CLI* on page 9) can enter console mode from the CLI and use the console commands. For details of how to use console commands, see *Console Access CLI commands* on page 55.

  There are two types of console command, and different access permissions exist for each type of command:

  - *Usable commands* - using these commands will not lead to inconsistencies between the Information Model and the underlying system.
  - *Blacklisted commands* - using blacklisted console commands can lead to inconsistencies between the Information Model and the underlying system and should be used **with extreme caution**.

2.2.1 Status of console commands

The appendices at the back of this guide describe the status of the console commands that were used in releases prior to 8.0. One of the following notes is included at the beginning of each console command description to explain the recommended usage of the console command with the CLI:

Note - This console command **has** been replaced by the CLI command <cross-reference to CLI command>. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

Each console command description also contains details of which CLI access permissions a user needs in order to use the command. For more information, see *Access permissions to the CLI* on page 9.

For more information about how to use the console commands in the CLI, refer to the *User Guide* that corresponds with your GlobespanVirata system.

2.2.2 Access permissions to the CLI

There are three access level options for CLI users:

- *default user* - can use CLI commands; cannot use usable console commands or blacklisted console commands.
- *engineer* - can use CLI commands and usable console commands; cannot use blacklisted console commands.
- *super user* - can use CLI commands, usable console commands and blacklisted console commands. Can also set up user login accounts, save backup configuration and restore factory settings.

2.3 CLI Command Groups

2.3.1 ISOS modules supported by the CLI

The ISOS modules which are supported by the CLI are listed below:

- Auto-provisioning administration, *autoprov*; see *Auto-provisioning CLI commands* on page 17.
- Bridge configuration, *bridge* module; see *Bridge CLI commands* on page 29.
- Console access; see *Console Access CLI commands* on page 55.
- DHCP Client configuration, *dhcpclient* module; see *DHCP Client CLI commands* on page 61.
- DHCP Relay configuration, *dhcprelay* module; see *DHCP Relay CLI commands* on page 101.
- DHCP Server configuration, *dhcpserver* module; see *DHCP Server CLI commands* on page 111.
- DNS Client configuration, *dnsclient* module; see *DNS Client CLI commands* on page 147.
- DNS Relay configuration, *dnsrelay* module, see *DNS Relay CLI commands* on page 157.
- Ethernet configuration, *ethernet* module; see *Ethernet CLI commands* on page 163.
- Firewall configuration, *firewall* module; see *Firewall CLI commands* on page 173.
- Frame Relay configuration, *framerelay* module; see *Frame Relay CLI commands* on page 223.
- IGMP configuration, *igmp* module; see *IGMP CLI commands* on page 239
- IPoA configuration, *ipoa* module; see *IPoA CLI commands* on page 245.
- MAC Spoofing, *macspoof* module; see *MAC Spoofing CLI commands* on page 271.
- NAT configuration, *nat* module; see *NAT CLI commands* on page 275.
- Port configuration, *port* module; see *Port CLI commands* on page 303.

- PPPoA configuration, *pppoa* module; see *PPPoA CLI commands* on page 311.
- PPPoE configuration, *pppoe* module; see *PPPoE CLI commands* on page 383.
- PPPoH configuration, pppoh module; see *PPPoH CLI commands* on page 451.
- PPTP configuration, *pptp* module; see *PPTP CLI commands* on page 501.
- Routing configuration - IP stack, *ip* module; see *TCP/IP CLI commands* on page 619.
- RFC1483 configuration, *rfc1483* module; see *RFC1483 CLI commands* on page 517.
- Security configuration, *security* module; see *Security CLI commands* on page 541.
- SNTP configuration, *sntp* module; see *SNTP CLI commands* on page 571.
- System configuration; *system* module; see *System CLI commands* on page 587.
- Transports configuration, *transports* module; see *Transports CLI commands* on page 679.
- User configuration, *user* module: see *User CLI commands* on page 687.
- WebServer administration, *webserver*; see *Web Server CLI commands* on page 693.

2.3.2 ISOS modules NOT supported by the CLI

The ISOS modules which are **not** supported by the CLI are listed below. For these modules, the Console interface can be used for configuration:

- BUN; see *BUN Console Commands* on page 747.
- TFTP; see *TFTP Console Commands* on page 891.
- Chips; see *Chips Console Commands* on page 771

There are other support modules that are not supported by the CLI, including:

buffer	oamcli
glenelg	portcli
flashfs	q93b
fm	snmp
ilmi	sscop
isfs	switchcli
led	switchctrl
oam	traffic

For information about the console commands for these ATMOS modules, refer to the appropriate *Functional Specification* for the module.

2.4 CLI Description

This section describes the conventions used in this manual for describing the CLI commands.

2.4.1 Understanding CLI/Console command tables

Each chapter of this manual describes a CLI command group. A chapter about an ISOS module that is supported by the CLI has two tables at the beginning of it:

- The first table in the chapter lists all the CLI commands available in that CLI command group.
- The second table contains the console commands used in releases before 8.0. The status of each console command is confirmed as one of the following:
  - replaced by a CLI command <link to further information on the CLI command>
  - usable <link to information on the console command in the relevant Appendix>
  - blacklisted <link to information on the console command in the relevant Appendix>

A chapter about an ISOS module that is not supported by the CLI only has the console commands table at the beginning of it. Similarly, if there are no console commands for a particular ISOS module, that chapter has no CLI command table.

A typical extract from a console/CLI command table is shown below:

Console command	Status of command in this release
event	Usable command, see *event …* on page 707
restart	Replaced by CLI command *restart* on page 48
tell	Blacklisted command, see *tell <process> …*

2.4.2 Using the Index

The Index chapter at the end of this manual provides you with a quick reference guide to all of the CLI and console commands, and tells you which category each command belongs to. For example:

console process (CLI) 60
cpu (Chips) (usable) 618
crlf, nocrlf (blacklisted) 571

2.4.3 Identification in the CLI

Existing CLI interfaces, transports, tunnels (and so on) can be identified by their name or number. You assign a name when you create the interface (or transport, tunnel etc) using the *add interface* command. Once created, the CLI adds this interface to a list of IP interfaces which you can display using the *ip list interfaces* command. Interfaces are given identification numbers that appear under the first column under the heading *ID*:

```
prompt> ip list interfaces
IP Interfaces:

 ID  |    Name      |    IP Address    |   DHCP   |   NAT    |   Transport
-----|--------------|------------------|----------|----------|-------------
   1 | ip4          | 0.0.0.0          | disabled | disabled | Not attached
   2 | ppp_device   | 192.168.102.2    | disabled | disabled | pppoe1
   3 | ip2          | 192.168.102.3    | disabled | disabled | eth1
   4 | ip1          | 0.0.0.0          | disabled | disabled | ipoa1
-------------------------------------------------------------------------
```

Identification numbers **only** reflect the order that they appear in the interface list. If you delete the IP interface called *ip2* with the identification number *3*, the IP interface *ip1* moves up the list and inherits the identification number *3*.

2.5 Adding new CLI commands

You can create CLI commands that configure and read values and attributes that you have defined. For information on how to do this, see the *ISOS Management Developer's Guide: DO-008640-PS*.

2.6 Using the source CLI command

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 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:

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

prompt> ip clear interfaces

prompt> ip clear routes

prompt> transports clear

prompt> ethernet add transport eth1 ethernet

prompt> ip add interface ip1 192.168.101.2

prompt> ip attach ip1 eth1

prompt> ipoa add transport ipoa1 pvc a1 0 700

prompt> ip add interface ip2 192.168.102.2

prompt> ip attach ip2 ipoa1

prompt> ipoa transport ipoa1 set pvc 1 pcr 50000

prompt> ip add route default 0.0.0.0 0.0.0.0 gateway 192.168.102.3

prompt> ip list interfaces
```

```
IP Interfaces:

 ID  |    Name      |    IP Address    |   DHCP   |   Transport
-----|--------------|------------------|----------|----------------
   1 | ip2          | 192.198.102.2    | disabled | ipoa1
   2 | ip1          | 192.168.101.2    | disabled | eth1
-------------------------------------------------------------------

prompt--> transports list

Services:

 ID  |    Name      | Type
-----|--------------|-------------------------------------------------------
   1 | ipoa1        | IPoA      | TxPkts:   19/0    RxPkts:    0/0
   2 | eth1         | Ethernet  | TxPkts:  152/0    RxPkts:  152/0
-------------------------------------------------------------------------------

IP routes:

prompt> ip list routes

IP routes:

 ID  |    Name    | Destination      | Netmask          | Gateway/Interface
-----|------------|------------------|------------------|------------------
   1 | default    | 0.0.0.0          | 0.0.0.0          | 192.168.102.3
-------------------------------------------------------------------------------
```

3. Auto-provisioning CLI commands

This chapter describes the auto-provisioning CLI commands.

3.1 Summary

3.1.1 Auto-provisioning CLI commands

The table below lists the *Auto-provisioning* commands provided by the CLI:

Command	Reference
autoprov disable	*autoprov disable* on page 19
autoprov autopvc	*autoprov autopvc* on page 20
autoprov opendsl	*autoprov opendsl* on page 21
autoprov set defterm	*autoprov set defterm* on page 22
autoprov set nat	*autoprov set nat* on page 23
autoprov set pollinterval	*autoprov set pollinterval* on page 24
autoprov set ppp-pass	*autoprov set ppp-pass* on page 25
autoprov set ppp-user	*autoprov set ppp-user* on page 26
autoprov show	*autoprov show* on page 27
autoprov show opendsl	*autoprov show* on page 27

3.1.2 Auto-provisioning Console commands

There are **no** console commands for the *autoprov* process.

3.2 autoprov disable

3.2.1 Syntax

```
autoprov disable
```

3.2.2 Description

This command disables the auto-provisioning protocol that you have built into your image. There are two types of auto-provisioning supported:

- OpenDSL
- TR037 AutoPVC

By default, the auto-provisioning process that is built into your image is enabled.

Each auto-provisioning protocol allows your device to query a management entity (typically a DSLAM) to obtain its local configuration. The device uses the information that it is given to provision the appropriate services.

3.2.3 Example

```
prompt> autoprov disable
```

3.2.4 See also

ATMOS OpenDSL Functional Specification: DO-008383-PS

TR037 AutoPVC Functional Specification: DO-008520-PS

3.3 autoprov autopvc

3.3.1 Syntax

```
autoprov autopvc
```

3.3.2 Description

This command enables the TR037 AutoPVC auto-provisioning protocol. You must have the TR037 AutoPVC process included in your image build in order to use this protocol.

3.3.3 Example

```
prompt> autoprov autopvc
```

3.3.4 See also

TR037 AutoPVC Functional Specification: DO-008520-PS

For information on including processes in your image build, see the Software User's Guide for the GlobespanVirata system that you are using.

3.4 autoprov opendsl

3.4.1 Syntax

```
autoprov opendsl
```

3.4.2 Description

This command enables the OpenDSL auto-provisioning protocol. You must have the OpenDSL process included in your image build in order to use this protocol.

3.4.3 Example

```
prompt> autoprov opendsl
```

3.4.4 See also

TR037 AutoPVC Functional Specification: DO-008520-PS

For information on including processes in your image build, see the Software User's Guide for the GlobespanVirata system that you are using.

3.5 autoprov set defterm

3.5.1 Syntax

```
autoprov set defterm {router|bridge|disabled}
```

3.5.2 Description

This command is only available if you are using the TR037 AutoPVC process.

This command sets the default termination method that TR037 AutoPVC uses for protocols that can be terminated either at the router or the bridge (e.g., PPP).

3.5.3 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
router	Terminates the PVCs of relevant protocols at the router.	bridge
bridge	Terminates the PVCs of relevant protocols at the bridge.	
disabled	No default termination method is set.	

3.5.4 Example

```
prompt> autoprov set defterm router
```

3.5.5 See also

TR037 AutoPVC Functional Specification: DO-008520-PS

3.6 autoprov set nat

3.6.1 Syntax

```
autoprov set nat {enabled|disabled}
```

3.6.2 Description

This command is only available if you are using the TR037 AutoPVC process.

This command specifies whether NAT is to be automatically enabled/disabled by the TR037 AutoPVC process on interfaces.

3.6.3 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
enabled	Sets TR037 AutoPVC to enable NAT on autoconfigured interfaces.	enabled
disabled	Sets TR037 AutoPVC to disable NAT on autoconfigured interfaces.	

3.6.4 Example

```
prompt> autoprov set nat enabled
```

3.6.5 See also

TR037 AutoPVC Functional Specification: DO-008520-PS

3.7 autoprov set pollinterval

3.7.1 Syntax

```
autoprov set pollinterval <interval>
```

3.7.2 Description

This command is only available if you are using the OpenDSL process.

Once the appropriate services have been provisioned, auto-provisioning polls the DSLAM at regular intervals to detect and implement any changes to the initial configuration. This command sets the length of time between poll intervals.

3.7.3 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
interval	The length of time in seconds that auto-provisioning waits since the last time it polled the DSLAM before polling the DSLAM again.	45

3.7.4 Example

```
prompt> autoprov set pollinterval 60
```

3.8 autoprov set ppp-pass

3.8.1 Syntax

```
autoprov set ppp-pass <password>
```

3.8.2 Description

This command is only available if you are using the TR037 AutoPVC process.

This command sets a password for PPP services.

3.8.3 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
password	A unique password made up of more than one character that identifies an individual user and lets the user access PPP services.	N/A

3.8.4 Example

```
prompt> autoprov set ppp-pass jupiter
```

3.8.5 See also

pppoa set transport password on page 358

pppoe set transport password on page 425

3.9 autoprov set ppp-user

3.9.1 Syntax

```
autoprov set ppp-user <name>
```

3.9.2 Description

This command is only available if you are using the TR037 AutoPVC process.

This command sets a username for PPP services.

3.9.3 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 unique login name made up of more than one character that identifies an individual user and lets the user access PPP services.	N/A

3.9.4 Example

```
prompt> autoprov set ppp-user ckearns
```

3.9.5 See also

pppoa set transport username on page 373

pppoe set transport username on page 442

3.10 autoprov show

3.10.1 Syntax

```
autoprov show
```

3.10.2 Description

This command displays information about the auto-provisioning process that you are using.

If you are using TR037 AutoPVC, the following information is displayed:

- enabled status (true or false)
- current state of the process (Idle, Fetching, Failed, Configured)
- the default termination method for a protocol that can be terminated either at the router or the bridge (Bridge, Router, Disabled).
- NAT enable status (**Important**: see the note for *autoprov set nat* on page 23)
- PPP username
- PPP password

If you are using OpenDSL, the following information is displayed:

- status (enabled or disabled)
- current state of the process
- poll interval set
- VCI set

3.10.3 Examples

Example One - TR037 AutoPVC

```
prompt> autoprov show
TR037 AutoPVC Configuration:

Enabled : true
State : Idle
Default Termination : Bridge
Enable Nat : false
```

```
Ppp Username : testuser
Ppp Password : testpass
```

Example Two - OpenDSL

```
prompt> show autoprov
OpenDSL Configuration:
Status : ENABLED

State : Unconfigured
Poll interval : 45
VCI : 16
```

3.10.4 See also

ATMOS OpenDSL Functional Specification: DO-008383-PS

TR037 AutoPVC Functional Specification: DO-008520-PS

4. Bridge CLI commands

This chapter describes the Bridge CLI commands.

4.1 Summary

4.1.1 Bridge CLI commands

The table below lists the Bridge commands provided by the CLI:

Command	Reference
bridge add interface	*bridge add interface* on page 33
bridge attach	*bridge attach* on page 34
bridge clear interfaces	*bridge clear interfaces* on page 35
bridge delete interface	*bridge delete interface* on page 36
bridge detach interface	*bridge detach interface* on page 37
bridge list interfaces	*bridge list interfaces* on page 38
bridge set filterage	*bridge set filterage* on page 39
bridge set interface	*bridge set interface filtertype* on page 40
bridge set spanning enabled/disabled	*bridge set spanning* on page 41
bridge set spanning forwarddelay	*bridge set spanning forwarddelay* on page 42
bridge set spanning hellotime	*bridge set spanning hellotime* on page 44
bridge set spanning maxage	*bridge set spanning maxage* on page 45
bridge set spanning priority	*bridge set spanning priority* on page 46
bridge show	*bridge show* on page 47
bridge show interface	*bridge show interface* on page 49

4.1.2 Bridge Console commands

The table below lists the *bridge* **console** commands and, if available, their CLI equivalent command:

Command	CLI Equivalent
bridge device add	Replaced by CLI command *bridge add interface* on page 33
bridge device delete	Replaced by CLI command *bridge delete interface* on page 36
bridge device list	Replaced by CLI command *bridge list interfaces* on page 38
bridge ethertype	Replaced by CLI command *bridge set interface filtertype* on page 40
bridge filter	Blacklisted command, see *filter* on page 735
bridge filterage	Replaced by CLI command *bridge set filterage* on page 39
bridge flush	Usable command, see *flush* on page 737
bridge help	Usable command
bridge info	Replaced by CLI command *bridge show* on page 47
bridge interface	Blacklisted command, see *interface* on page 739
bridge portfilter	Blacklisted command, see *portfilter* on page 740
bridge spanning enable/disable	Replaced by CLI command *bridge set spanning* on page 41
bridge spanning forwarddelay	Replaced by CLI command *bridge set spanning forwarddelay* on page 42
bridge spanning hellotime	Replaced by CLI command *bridge set spanning hellotime* on page 44
bridge spanning info	Replaced by CLI command *bridge show* on page 47
bridge spanning event	Usable command, see *spanning* on page 742
bridge spanning maxage	Replaced by CLI command *bridge set spanning maxage* on page 45

Command	CLI Equivalent
bridge spanning port	Blacklisted command, see *spanning* on page 742
bridge spanning priority	Replaced by CLI command *bridge set spanning priority* on page 46
bridge spanning status	Replaced by CLI command *bridge show* on page 47
bridge spanning version	Replaced by CLI command *bridge show* on page 47
bridge status	Replaced by CLI command *bridge show* on page 47 and *bridge list interfaces* on page 38
bridge version	Replaced by CLI command *bridge show* on page 47

4.2 bridge add interface

4.2.1 Syntax

```
bridge add interface <name>
```

4.2.2 Description

This command adds a named interface to the bridge.

4.2.3 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	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

4.2.4 Example

```
prompt> bridge add interface bridge1
```

4.2.5 See also

bridge attach on page 34

bridge list interfaces on page 38

For information on creating and attaching interfaces and transports, see the Software User's Guide for the GlobespanVirata system that you are using.

4.3 bridge attach

4.3.1 Syntax

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

4.3.2 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.

4.3.3 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 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

4.3.4 Example

```
prompt> bridge attach bridge1 my1483
```

4.3.5 See also

bridge add interface on page 33

bridge list interfaces on page 38

4.4 bridge clear interfaces

4.4.1 Syntax

```
bridge clear interfaces
```

4.4.2 Description

This command deletes all bridge interfaces that were created using the *bridge add interfac*e command.

4.4.3 Example

```
prompt> bridge clear interfaces
```

4.4.4 See also

bridge delete interface on page 36

4.5 bridge delete interface

4.5.1 Syntax

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

4.5.2 Description

This command deletes a single interface from the bridge configuration.

4.5.3 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 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

4.5.4 Example

```
prompt> bridge delete interface 1
```

4.5.5 See also

bridge list interfaces on page 38

4.6 bridge detach interface

4.6.1 Syntax

```
bridge detach interface {<name>|<number>}
```

4.6.2 Description

This command detaches the transport that was attached to the bridge interface using the *bridge attach interface* command.

4.6.3 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 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

4.6.4 Example

```
prompt> bridge detach interface 2
```

4.6.5 See also

bridge list interfaces on page 38

4.7 bridge list interfaces

4.7.1 Syntax

```
bridge list interfaces
```

4.7.2 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)

4.7.3 Example

```
prompt> bridge list interfaces
Bridge Interfaces:

 ID  |   Name     | Filter Type |    Transport
-----|------------|-------------|------------------
   1 | bridge3    | All         | eth1
   2 | bridge2    | All         | Not attached
   3 | bridge1    | All         | Not attached
-------------------------------------------------
```

4.7.4 See also

bridge set filterage on page 39

bridge set interface filtertype on page 40

4.8 bridge set filterage

4.8.1 Syntax

```
bridge set filterage <filter age>
```

4.8.2 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.

4.8.3 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
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

4.8.4 Example

```
prompt> bridge set filterage 2000
```

4.8.5 See also

bridge show interface on page 49

4.9 bridge set interface filtertype

4.9.1 Syntax

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

4.9.2 Description

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

4.9.3 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 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.	

4.9.4 Example

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

4.9.5 See also

bridge list interfaces on page 38

4.10 bridge set spanning

4.10.1 Syntax

```
bridge set spanning { enabled | disabled }
```

4.10.2 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.

4.10.3 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
enabled	Allows the bridge to use the spanning tree protocol.	disabled
disabled	Ensures that the bridge acts as a transparent bridge.	

4.10.4 Example

```
prompt> bridge set spanning enabled
```

4.10.5 See also

bridge set spanning forwarddelay on page 42

bridge set spanning hellotime on page 44

bridge set spanning maxage on page 45

bridge set spanning priority on page 46

bridge show on page 47

For more information on the spanning tree protocol, see the *ATMOS Spanning Tree Specification: DO-007085-PS*.

4.11 bridge set spanning forwarddelay

4.11.1 Syntax

```
bridge set spanning forwarddelay <delay>
```

4.11.2 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.

4.11.3 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
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

4.11.4 Example

```
prompt> bridge set spanning forwarddelay 20
```

4.11.5 See also

bridge set spanning on page 41

bridge set spanning hellotime on page 44

bridge set spanning maxage on page 45

bridge set spanning priority on page 46

bridge show on page 47

For more information on the spanning tree protocol, see the *ATMOS Spanning Tree Specification: DO-007085-PS*.

4.12 bridge set spanning hellotime

4.12.1 Syntax

```
bridge set spanning hellotime <hellotime>
```

4.12.2 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.

4.12.3 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
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* on page 41.	2

4.12.4 Example

```
prompt> bridge set spanning hellotime 10
```

4.12.5 See also

bridge set spanning on page 41

bridge set spanning forwarddelay on page 42

bridge set spanning maxage on page 45

bridge set spanning priority on page 46

bridge show on page 47

For more information on the spanning tree protocol, see the *ATMOS Spanning Tree Specification: DO-007085-PS*.

4.13 bridge set spanning maxage

4.13.1 Syntax

```
bridge set spanning maxage <maxage>
```

4.13.2 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.

4.13.3 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
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* on page 41.	20

4.13.4 Example

```
prompt> bridge set spanning maxage 30
```

4.13.5 See also

bridge set spanning on page 41

bridge set spanning forwarddelay on page 42

bridge set spanning hellotime on page 44

bridge set spanning priority on page 46

bridge show on page 47

For more information on the spanning tree protocol, see the *ATMOS Spanning Tree Specification: DO-007085-PS*.

4.14 bridge set spanning priority

4.14.1 Syntax

```
bridge set spanning priority <priority>
```

4.14.2 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.

4.14.3 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
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

4.14.4 Example

```
prompt> bridge set spanning priority 1000
```

4.14.5 See also

bridge set spanning on page 41

bridge set spanning forwarddelay on page 42

bridge set spanning hellotime on page 44

bridge set spanning maxage on page 45

bridge show on page 47

For more information on the spanning tree protocol, see the *ATMOS Spanning Tree Specification: DO-007085-PS*.

4.15 bridge show

4.15.1 Syntax

```
bridge show
```

4.15.2 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)

4.15.3 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
```

4.15.4 See also

bridge set spanning on page 41

bridge set spanning forwarddelay on page 42

bridge set spanning hellotime on page 44

bridge set spanning maxage on page 45

For more information on the spanning tree protocol, see the *ATMOS Spanning Tree Specification: DO-007085-PS*.

4.16 bridge show interface

4.16.1 Syntax

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

4.16.2 Description

This command displays the filter type value of a named bridge interface.

Note - This command **does not** shows the current contents of the bridge's filter table. See the console command *filter* on page 735.

4.16.3 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 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

4.16.4 Example

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

4.16.5 See also

bridge set interface filtertype on page 40

5. BUN CLI commands

This chapter describes CLI support for BUN.

5.1 Summary

5.1.1 BUN CLI commands

There is currently no support for BUN in the CLI.

5.1.2 BUN Console commands

The table below lists the *bun* **console** commands and, if they are usable or black-listed:

Command	CLI Equivalent
help	Usable command, see *help* on page 751
version	Usable command, see *version* on page 752
build	Usable command, see *build* on page 753
config	Blacklisted command, see *config* on page 754
list config	Usable command, see *list config* on page 755
list devices	Usable command, see *list devices* on page 757
show device	Usable command, see *show device* on page 758
list classes	Usable command, see *list classes* on page 760
show class	Usable command, see *show class* on page 761
list ports	Usable command, see *list ports* on page 762
show port	Usable command, see *show port* on page 763
set port	Blacklisted command, see *set port* on page 764
list channels	Usable command, see *list channels* on page 765
list all open channels	Usable command, see *list all open channels* on page 766
show channel	Usable command, see *show channel* on page 767
set channel	Blacklisted command, see *set channel* on page 768
reset port	Blacklisted command, see *reset port* on page 769

6. Chips CLI commands

This chapter describes CLI support for the Chips module.

6.1 Summary

6.1.1 Chips CLI commands

There is currently no support for Chips in the CLI.

6.1.2 Chips Console commands

The table below lists the *chips* **console** commands and, if they are usable or black-listed:

Command	CLI Equivalent
chips cpu	Usable command, see *cpu* on page 772
chips info	Usable command, see *info* on page 776
chips mem	Usable command, see *mem* on page 777
chips help	Usable command, see *help* on page 775
chips debug	Blacklisted command, see *debug* on page 773
chips exit	Blacklisted command, see *exit* on page 774
chips rb, rh, rw, wb, wh, ww	Blacklisted command see *rb, rh, rw, wb, wh, ww* on page 778
chips steal	Blacklisted command, see *steal* on page 780
chips tell	Blacklisted command, see *tell* on page 781

7. Console Access CLI commands

This chapter describes the console access CLI commands.

7.1 Summary

7.1.1 Console access CLI commands

The table below lists the *console access* commands provided by the CLI:

Command	Description/Console command
console enable	*console enable* on page 57
console process	*console process* on page 58

7.1.2 Console Commands

The table below lists the *access* console commands*:*

Command	CLI status
exit	Blacklisted command, see *Console command - exit* on page 59

7.2 console enable

7.2.1 Syntax

```
console enable
```

7.2.2 Description

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

7.2.3 Example

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

7.2.4 See also

console process on page 58 and *Console command - exit* on page 59.

7.3 console process

7.3.1 Syntax

```
console process <console command>
```

7.3.2 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 Super user access can use this command.

7.3.3 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
console command	A **usable** console command. You can find a list of usable commands with a link to further information about each usable command at the start of each chapter in this manual.	N/A

7.3.4 Example

The following *console process* example enters the *usable* console command, *bridge portfilter*:

```
prompt> console process bridge portfilter
portfilter 2 all
portfilter 3 all
```

7.3.5 See also

console enable on page 57 if you have Super user access and want to enter more than one console command or use blacklisted commands.

For information on usable and blacklisted commands, see *Status of console commands* on page 9.

The console commands that you can use with the *console process* command are listed in *atmos/products/include/flashfs/cliconsole*.

7.4 Console command - exit

Note - This console command **has not** been replaced by a CLI command. This is a special console command to allow Super users to return to the CLI from the console.

7.4.1 Syntax

```
exit
```

7.4.2 Description

This console command allows you to return to the CLI after you have entered console mode using the command *console enable* on page 57. When you want to exit console mode and return to the CLI, you need to type `exit` in the *root* of the console.

Only Super users can use this command.

7.4.3 Example

```
prompt> exit
Returning to CLI from console
```

7.4.4 See also

console enable on page 57.

8. DHCP Client CLI commands

This chapter describes the DHCP client CLI commands.

8.1 Summary

8.1.1 DHCP client CLI commands

The table below lists the DHCP client commands provided by the CLI:

Command	Reference
dhcpclient add interfaceconfig	*dhcpclient add interfaceconfig* on page 65
dhcpclient clear interfaceconfigs	*dhcpclient clear interfaceconfigs* on page 66
dhcpclient delete interfaceconfig	*dhcpclient delete interfaceconfig* on page 67
dhcpclient interfaceconfig add requested option	*dhcpclient interfaceconfig add requested option* on page 68
dhcpclient interfaceconfig add required option	*dhcpclient interfaceconfig add required option* on page 70
dhcpclient interfaceconfig add sent option	*dhcpclient interfaceconfig add sent option* on page 72
dhcpclient interfaceconfig clear sent options	*dhcpclient interfaceconfig clear sent options* on page 74
dhcpclient interfaceconfig clear requested options	*dhcpclient interfaceconfig clear requested options* on page 75
dhcpclient interfaceconfig delete requested option	*dhcpclient interfaceconfig delete requested option* on page 76
dhcpclient interfaceconfig delete sent option	*dhcpclient interfaceconfig delete sent option* on page 77
dhcpclient interfaceconfig list requested options	*dhcpclient interfaceconfig list requested options* on page 78
dhcpclient interfaceconfig list sent options	*dhcpclient interfaceconfig list sent options* on page 80
dhcpclient list interfaceconfigs	*dhcpclient list interfaceconfigs* on page 82
dhcpclient set backoff	*dhcpclient set backoff* on page 83

Command	Reference
dhcpclient set interfaceconfig autoip	*dhcpclient set interfaceconfig autoip enabled|disabled* on page 84
dhcpclient set interfaceconfig clientid	*dhcpclient set interfaceconfig clientid* on page 86
dhcpclient set interfaceconfig defaultroute enabled|disabled	*dhcpclient set interfaceconfig defaultroute enabled|disabled* on page 87
dhcpclient interfaceconfig set dhcpserverpoolsize	*dhcpclient set interfaceconfig dhcpserverpoolsize* on page 88
dhcpclient interfaceconfig set dhcpserverinterface	*dhcpclient set interfaceconfig dhcpserverinterface* on page 90
dhcpclient set interfaceconfig givednstoclient enabled|disabled	*dhcpclient set interfaceconfig givednstoclient enabled|disabled* on page 91
dhcpclient set interfaceconfig givednstorelay enabled|disabled	*dhcpclient set interfaceconfig givednstorelay enabled|disabled* on page 92
dhcpclient set interfaceconfig interface	*dhcpclient set interfaceconfig interface* on page 93
dhcpclient set interfaceconfig noclientid	*dhcpclient set interfaceconfig noclientid* on page 94
dhcpclient set interfaceconfig requestedleasetime	*dhcpclient set interfaceconfig requestedleasetime* on page 95
dhcpclient set reboot	*dhcpclient set reboot* on page 96
dhcpclient set retry	*dhcpclient set retry* on page 97
dhcpclient show	*dhcpclient show* on page 98
dhcpclient update	*dhcpclient update* on page 99

8.1.2 DHCP client Console commands

The table below lists the *dhcp-client* **console** commands and, if available, their equivalent CLI command:

Command	CLI Equivalent
dhcpclient config	Replaced by CLI command *dhcpclient show* on page 98
dhcpclient help	Usable command, see *help* on page 785
dhcpclient pool	Usable command, see *pool* on page 786
dhcpclient status	Usable command, see *status* on page 787
dhcpclient trace	Usable command, see *trace* on page 788

8.1.3 DHCP client default settings

By default, DHCP client requests the following information from DHCP server:

- the DHCP server IP address and subnet mask
- DNS server addresses. By default, received DNS server addresses are passed on to the DNS relay and not passed to DNS client. To change these default settings, use the commands *dhcpclient set interfaceconfig givednstoclient enabled|disabled* on page 91 and *dhcpclient set interfaceconfig givednstorelay enabled|disabled* on page 92.
- DHCP server default gateway information. By default, DHCP client makes use of default gateway information. To change this default setting, use the command *dhcpclient set interfaceconfig defaultroute enabled|disabled* on page 87.

8.2 dhcpclient add interfaceconfig

8.2.1 Syntax

```
dhcpclient add interfaceconfig <name> <ipinterface>
```

8.2.2 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.

8.2.3 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	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

8.2.4 Example

```
prompt> dhcpclient add interfaceconfig config1 ip1
```

8.2.5 See also

dhcpclient list interfaceconfigs on page 82

ip list interfaces on page 648

ip set interface dhcp on page 656

8.3 dhcpclient clear interfaceconfigs

8.3.1 Syntax

```
dhcpclient clear interfaceconfigs
```

8.3.2 Description

This command deletes all existing DHCP client interface configurations.

8.3.3 Example

```
prompt> dhcpclient clear interfaceconfigs
```

8.3.4 See also

dhcpclient list interfaceconfigs on page 82

8.4 dhcpclient delete interfaceconfig

8.4.1 Syntax

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

8.4.2 Description

This command deletes a single DHCP client interface configuration.

8.4.3 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 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

8.4.4 Example

```
prompt> dhcpclient delete interfaceconfig config1
```

8.4.5 See also

dhcpclient list interfaceconfigs on page 82

8.5 dhcpclient interfaceconfig add requested option

8.5.1 Syntax

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

8.5.2 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, and a list of available option names is available in *DHCP Server Functional Specification: DO-007343-PS.*

8.5.3 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 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

8.5.4 Example

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

8.5.5 See also

dhcpclient list interfaceconfigs on page 82

dhcpclient interfaceconfig add requested option on page 68

dhcpclient interfaceconfig add required option on page 70

For information on RFC 2132, see *http://www.ietf.org/rfc/rfc2132.txt*

DHCP Server Functional Specification: DO-007343-PS

8.6 dhcpclient interfaceconfig add required option

8.6.1 Syntax

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

8.6.2 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, and a list of available option names is available in *DHCP Server Functional Specification: DO-007343-PS.*

8.6.3 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 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

8.6.4 Example

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

8.6.5 See also

dhcpclient list interfaceconfigs on page 82

dhcpclient interfaceconfig add requested option on page 68

dhcpclient interfaceconfig add required option on page 70

For information on RFC 2132, see *http://www.ietf.org/rfc/rfc2132.txt*

DHCP Server Functional Specification: DO-007343-PS

8.7 dhcpclient interfaceconfig add sent option

8.7.1 Syntax

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

8.7.2 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.

8.7.3 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 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

8.7.4 Example

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

This command example tells the DHCP client to send the DHCP host-name 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. For more information, see the *Known Issues and Troubleshooting* chapter of the *DHCP Server Functional Specification*.

8.7.5 See also

dhcpclient list interfaceconfigs on page 82

dhcpclient interfaceconfig list sent options on page 80

For information on RFC 2132, see *http://www.ietf.org/rfc/rfc2132.txt*

DHCP Server Functional Specification: DO-007343-PS

8.8 dhcpclient interfaceconfig clear sent options

8.8.1 Syntax

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

8.8.2 Description

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

8.8.3 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
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

8.8.4 Example

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

8.8.5 See also

dhcpclient list interfaceconfigs on page 82

dhcpclient interfaceconfig list sent options on page 80

dhcpclient interfaceconfig add sent option on page 72

dhcpclient interfaceconfig delete sent option on page 77

For information on RFC 2132, see *http://www.ietf.org/rfc/rfc2132.txt*

8.9 dhcpclient interfaceconfig clear requested options

8.9.1 Syntax

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

8.9.2 Description

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

8.9.3 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 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

8.9.4 Example

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

8.9.5 See also

dhcpclient list interfaceconfigs on page 82

dhcpclient interfaceconfig add requested option on page 68

dhcpclient interfaceconfig add required option on page 70

dhcpclient interfaceconfig delete requested option on page 76

8.10 dhcpclient interfaceconfig delete requested option

8.10.1 Syntax

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

8.10.2 Description

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

8.10.3 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 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

8.10.4 Example

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

8.10.5 See also

dhcpclient list interfaceconfigs on page 82

dhcpclient interfaceconfig add requested option on page 68

dhcpclient interfaceconfig add required option on page 70

dhcpclient interfaceconfig clear requested options on page 75

8.11 dhcpclient interfaceconfig delete sent option

8.11.1 Syntax

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

8.11.2 Description

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

8.11.3 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 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

8.11.4 Example

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

8.11.5 See also

dhcpclient list interfaceconfigs on page 82

dhcpclient interfaceconfig list sent options on page 80

dhcpclient interfaceconfig add sent option on page 72

dhcpclient interfaceconfig clear sent options on page 74

8.12 dhcpclient interfaceconfig list requested options

8.12.1 Syntax

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

8.12.2 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, and a list of available option names is available in *DHCP Server Functional Specification: DO-007343-PS.*

8.12.3 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 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

8.12.4 Example

```
prompt> dhcpclient interfaceconfig client1 list requested options

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

8.12.5 See also

dhcpclient interfaceconfig add requested option on page 68

dhcpclient interfaceconfig add required option on page 70

dhcpserver subnet add option on page 137

For information on RFC 2132, see *http://www.ietf.org/rfc/rfc2132.txt*

DHCP Server Functional Specification: DO-007343-PS

8.13 dhcpclient interfaceconfig list sent options

8.13.1 Syntax

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

8.13.2 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, and a list of available option names is available in *DHCP Server Functional Specification: DO-007343-PS.*

8.13.3 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 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

8.13.4 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         | globespanvirata
-------------------------------------------------
```

8.13.5 See also

dhcpclient interfaceconfig add sent option on page 72

dhcpclient interfaceconfig clear sent options on page 74

dhcpclient interfaceconfig delete sent option on page 77

dhcpserver subnet add option on page 137

For information on RFC 2132, see *http://www.ietf.org/rfc/rfc2132.txt*

DHCP Server Functional Specification: DO-007343-PS

8.14 dhcpclient list interfaceconfigs

8.14.1 Syntax

```
dhcpclient list interfaceconfigs
```

8.14.2 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)

8.14.3 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
-------------------------------------------------------------------|--------
```

8.14.4 See also

dhcpclient show on page 98

dhcpclient set interfaceconfig requestedleasetime on page 95

dhcpclient set interfaceconfig clientid on page 86

dhcpclient set interfaceconfig autoip enabled|disabled on page 84

8.15 dhcpclient set backoff

8.15.1 Syntax

```
dhcpclient set backoff <backofftime>
```

8.15.2 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.

8.15.3 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
backofftime	The maximum number of seconds that the DHCP client can pause for between unsuccessful DHCP negotiations.	120

8.15.4 Example

```
prompt> dhcpclient set backoff 200
```

8.15.5 See also

dhcpclient show on page 98

8.16 dhcpclient set interfaceconfig autoip enabled|disabled

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

8.16.1 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.

Note - 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.

8.16.2 Options

The following table gives 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.	

8.16.3 Example

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

8.16.4 See also

dhcpserver subnet add option on page 137 (see the specific example given for this command)

For details of the GlobespanVirata implementation of IP address auto-configuration, see *DHCP Client Functional Specification: DO-007309-PS*.

For further information on the RFC standard for DHCP IP address auto-configuration, see *http://www.ietf.org/rfc/rfc2563.txt*.

8.17 dhcpclient set interfaceconfig clientid

8.17.1 Syntax

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

8.17.2 Description

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

8.17.3 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 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

8.17.4 Example

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

8.17.5 See also

dhcpclient list interfaceconfigs on page 82

8.18 dhcpclient set interfaceconfig defaultroute enabled|disabled

8.18.1 Syntax

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

8.18.2 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.

8.18.3 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 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.	

8.18.4 Example

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

8.18.5 See also

dhcpclient list interfaceconfigs on page 82

8.19 dhcpclient set interfaceconfig dhcpserverpoolsize

8.19.1 Syntax

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

8.19.2 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.

8.19.3 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 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

8.19.4 Example

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

8.19.5 See also

dhcpclient list interfaceconfigs on page 82

8.20 dhcpclient set interfaceconfig dhcpserverinterface

8.20.1 Syntax

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

8.20.2 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.

8.20.3 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 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

8.20.4 Example

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

8.20.5 See also

dhcpclient list interfaceconfigs on page 82

dhcpclient set interfaceconfig dhcpserverpoolsize on page 88

ip list interfaces on page 648

8.21 dhcpclient set interfaceconfig givednstoclient enabled|disabled

8.21.1 Syntax

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

8.21.2 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.

8.21.3 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 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.	

8.21.4 Example

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

8.21.5 See also

dhcpclient list interfaceconfigs on page 82

8.22 dhcpclient set interfaceconfig givednstorelay enabled|disabled

8.22.1 Syntax

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

8.22.2 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.

8.22.3 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 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.	

8.22.4 Example

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

8.22.5 See also

dhcpclient list interfaceconfigs on page 82

8.23 dhcpclient set interfaceconfig interface

8.23.1 Syntax

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

8.23.2 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.

8.23.3 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 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

8.23.4 Example

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

8.23.5 See also

dhcpclient list interfaceconfigs on page 82

ip list interfaces on page 648

ip set interface dhcp on page 656

8.24 dhcpclient set interfaceconfig noclientid

8.24.1 Syntax

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

8.24.2 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.

8.24.3 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 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

8.24.4 Example

```
prompt> dhcpclient set interfaceconfig client1 noclientid
```

8.24.5 See also

dhcpclient set interfaceconfig clientid on page 86

dhcpserver set allowunknownclients on page 124

8.25 dhcpclient set interfaceconfig requestedleasetime

8.25.1 Syntax

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

8.25.2 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.

8.25.3 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 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

8.25.4 Example

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

8.25.5 See also

dhcpclient list interfaceconfigs on page 82

dhcpserver set maxleasetime on page 127

dhcpserver set defaultleasetime on page 126

8.26 dhcpclient set reboot

8.26.1 Syntax

```
dhcpclient set reboot <reboottime>
```

8.26.2 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.

8.26.3 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
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

8.26.4 Example

```
prompt> dhcpclient set reboot 5
```

8.27 dhcpclient set retry

8.27.1 Syntax

```
dhcpclient set retry <retrytime>
```

8.27.2 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.

8.27.3 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
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

8.27.4 Example

```
prompt> dhcpclient set retry 150
```

8.28 dhcpclient show

8.28.1 Syntax

```
dhcpclient show
```

8.28.2 Description

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

- reboot time
- retry time
- maximum backoff time

8.28.3 Example

```
prompt> dhcpclient show
Global DHCP Client Configuration:


        Reboot time: 10
         Retry time: 300
  Max. backoff time: 120
```

8.28.4 See also

dhcpclient set reboot on page 96

dhcpclient set retry on page 97

dhcpclient set backoff on page 83

8.29 dhcpclient update

8.29.1 Syntax

```
dhcpclient update
```

8.29.2 Description

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

8.29.3 Example

```
prompt> dhcpclient update

dhcpclient: Reset request acknowledged. Reset imminent.
```

9. DHCP Relay CLI commands

This chapter describes the DHCP Relay CLI commands.

9.1 Summary

9.1.1 DHCP relay CLI commands

The table below lists the DHCP relay commands provided by the CLI:

Command	Description/Console command
dhcprelay add server	*dhcprelay add server* on page 103
dhcprelay clear servers	*dhcprelay delete server* on page 105
dhcprelay delete server	*dhcprelay delete server* on page 105
dhcprelay enable|disable	*dhcprelay enable|disable* on page 106
dhcprelay list servers	*dhcprelay list servers* on page 107
dhcprelay show	*dhcprelay show* on page 108
dhcprelay update	*dhcprelay update* on page 109

9.1.2 DHCP-relay Console commands

The table below lists the *dhcp relay* **console** commands and, if available, their equivalent CLI command:

Command	CLI Equivalent
dhcprelay add	Replaced by CLI command *dhcprelay add server* on page 103
dhcprelay config	Replaced by CLI command *dhcprelay list servers* on page 107
dhcprelay delete	Replaced by CLI command *dhcprelay delete server* on page 105
dhcprelay help	Blacklisted command, see *dhcprelay help* on page 795
dhcprelay pool	Blacklisted command, see *dhcprelay pool* on page 796
dhcprelay status	Blacklisted command, see *dhcprelay status* on page 797
dhcprelay trace/untrace	Blacklisted command, see *system log* on page 606
dhcprelay version	Blacklisted command, see *dhcprelay version* on page 800

9.2 dhcprelay add server

9.2.1 Syntax

```
dhcprelay add server <ipaddress>
```

9.2.2 Description

This command adds the IP address of a DHCP server subnet to DHCP relay's list of server IP addresses. The relay can store a maximum of 10 DHCP server addresses. Records of new IP addresses added are not updated until the *dhcprelay update* command has been entered.

9.2.3 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
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

9.2.4 Example

```
prompt> dhcprelay add server 239.252.197.0
```

9.2.5 See also

dhcpserver list subnets on page 123

dhcprelay update on page 109

9.3 dhcprelay clear servers

9.3.1 Syntax

```
dhcprelay clear servers
```

9.3.2 Description

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

9.3.3 Example

```
prompt> dhcprelay clear servers
```

9.3.4 See also

dhcprelay delete server on page 105

9.4 dhcprelay delete server

9.4.1 Syntax

```
dhcprelay delete server <number>
```

9.4.2 Description

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

9.4.3 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
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

9.4.4 Example

```
prompt> dhcprelay delete server 3
```

9.4.5 See also

dhcprelay list servers on page 107

dhcprelay clear servers on page 104

9.5 dhcprelay enable|disable

9.5.1 Syntax

```
dhcprelay {enable|disable}
```

9.5.2 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.

9.5.3 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
enable	Enables configuration of DHCP relay.	enable
disable	Disables configuration of DHCP relay.	

9.5.4 Example

```
prompt> dhcprelay enable
```

9.5.5 See also

dhcpserver enable|disable on page 119

9.6 dhcprelay list servers

9.6.1 Syntax

```
dhcprelay list servers
```

9.6.2 Description

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

9.6.3 Example

```
prompt> dhcprelay list servers
DHCP Servers:

 ID  | IP Address
-----|-----------------
   1 | 192.168.102.3
   2 | 239.252.197.0
------------------------
```

9.6.4 See also

dhcpserver list subnets on page 123

9.7 dhcprelay show

9.7.1 Syntax

```
dhcprelay show
```

9.7.2 Description

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

9.7.3 Example

```
prompt> dhcprelay show server
Global DHCP Relay Configuration:

                    Status: ENABLED
```

9.7.4 See also

dhcprelay enable|disable on page 106

9.8 dhcprelay update

9.8.1 Syntax

```
dhcprelay update
```

9.8.2 Description

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

9.8.3 Example

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

10.DHCP Server CLI commands

This chapter describes the DHCP server CLI commands.

10.1 Summary

10.1.1 DHCP server CLI commands

The table below lists the DHCP server commands provided by the CLI:

Command	Reference
dhcpserver add subnet	*dhcpserver add subnet* on page 115
dhcpserver clear subnets	*dhcpserver clear subnets* on page 117
dhcpserver delete subnet	*dhcpserver delete subnet* on page 118
dhcpserver enable|disable	*dhcpserver enable|disable* on page 119
dhcpserver list options	*dhcpserver list options* on page 120
dhcpserver list subnets	*dhcpserver list subnets* on page 123
dhcpserver set allowunknownclients	*dhcpserver set allowunknownclients* on page 124
dhcpserver set bootp	*dhcpserver set bootp* on page 125
dhcpserver set defaultleasetime	*dhcpserver set defaultleasetime* on page 126
dhcpserver set maxleasetime	*dhcpserver set maxleasetime* on page 127
dhcpserver set subnet defaultleasetime	*dhcpserver set subnet defaultleasetime* on page 128
dhcpserver set subnet hostisdefaultgateway	*dhcpserver set subnet hostisdefaultgateway* on page 129
dhcpserver set subnet hostisdnsserver	*dhcpserver set subnet hostisdnsserver* on page 130
dhcpserver set subnet maxleasetime	*dhcpserver set subnet maxleasetime* on page 131

Command	Reference
dhcpserver set subnet subnet	*dhcpserver set subnet subnet* on page 132
dhcpserver show	*dhcpserver show* on page 133
dhcpserver show subnet	*dhcpserver show subnet* on page 134
dhcpserver subnet add iprange	*dhcpserver subnet add iprange* on page 136
dhcpserver subnet add option	*dhcpserver subnet add option* on page 137
dhcpserver subnet clear ipranges	*dhcpserver subnet clear ipranges* on page 139
dhcpserver subnet clear options	*dhcpserver subnet clear options* on page 140
dhcpserver subnet delete iprange	*dhcpserver subnet delete iprange* on page 141
dhcpserver subnet delete option	*dhcpserver subnet delete option* on page 142
dhcpserver subnet list ipranges	*dhcpserver subnet list ipranges* on page 143
dhcpserver subnet list options	*dhcpserver subnet list options* on page 144
dhcpserver update	*dhcpserver update* on page 145

10.1.2 DHCP server Console commands

The table below lists the *dhcp server* **console** commands and, if available, their equivalent CLI command:

Command	CLI Equivalent
dhcpserver config	Replaced by CLI command *dhcpserver show* on page 133
dhcpserver help	Usable command, see *help* on page 804
dhcpserver pool	Usable command, see *pool* on page 805

Command	CLI Equivalent
dhcpserver reset	Blacklisted command, see *reset* on page 806
dhcpserver status	Usable command, see *status* on page 807
dhcpserver trace	Usable command, see *trace* on page 808
dhcpserver version	Usable command, see *version* on page 810

10.2 dhcpserver add subnet

10.2.1 Syntax

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

10.2.2 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.

10.2.3 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	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

10.2.4 Example

```
prompt>dhcpserver add subnet sub1 239.252.197.0 255.255.255.0
239.252.197.10 239.252.197.107
```

10.2.5 See also

dhcpserver list subnets on page 123

10.3 dhcpserver clear subnets

10.3.1 Syntax

```
dhcpserver clear subnets
```

10.3.2 Description

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

10.3.3 Example

```
prompt> dhcpserver clear subnets
```

10.3.4 See also

dhcpserver delete subnet on page 118

10.4 dhcpserver delete subnet

10.4.1 Syntax

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

10.4.2 Description

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

10.4.3 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 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

10.4.4 Example

```
prompt> dhcpserver delete subnet sub1
```

10.4.5 See also

dhcpserver clear subnets on page 117

10.5 dhcpserver enable|disable

10.5.1 Syntax

```
dhcpserver {enable|disable}
```

10.5.2 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.

10.5.3 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
enable	Enables configuration of the DHCP server.	enable
disable	Disables configuration of the DHCP server.	

10.5.4 Example

```
prompt> dhcpserver enable
```

10.5.5 See also

dhcprelay enable|disable on page 106

10.6 dhcpserver list options

10.6.1 Syntax

```
dhcpserver list options
```

10.6.2 Description

This command lists the option data types available for DHCP server. These options are detailed in RFC2132. The list is also available in the *DHCP Server Functional Specification: DO-007343-PS*.

You can configure the DHCP server using any of the options listed.

10.6.3 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
```

10.6.4 See also

dhcpserver subnet add option on page 137

For information on RFC 2132, see *http://www.ietf.org/rfc/rfc2132.txt*

DHCP Server Functional Specification: DO-007343-PS

10.7 dhcpserver list subnets

10.7.1 Syntax

```
dhcpserver list subnets
```

10.7.2 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)

10.7.3 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
-------------------------------------------------------------------------------
```

10.7.4 See also

dhcpserver show subnet on page 134

10.8 dhcpserver set allowunknownclients

10.8.1 Syntax

```
dhcpserver set allowunknownclients {enabled|disabled}
```

10.8.2 Description

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

10.8.3 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
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.	

10.8.4 Example

```
prompt> dhcpserver set allowunknownclients disabled
```

10.8.5 See also

dhcpclient set interfaceconfig clientid on page 86

10.9 dhcpserver set bootp

10.9.1 Syntax

```
dhcpserver set bootp {enabled|disabled}
```

10.9.2 Description

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

10.9.3 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
enabled	DHCP server responds to BOOTP queries.	enabled
disabled	DHCP server does not respond to BOOTP queries.	

10.9.4 Example

```
prompt> dhcpserver set bootp disabled
```

10.10 dhcpserver set defaultleasetime

10.10.1 Syntax

```
dhcpserver set defaultleasetime <defaultleasetime>
```

10.10.2 Description

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

10.10.3 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
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

10.10.4 Example

```
prompt> dhcpserver set defaultleasetime 50000
```

10.10.5 See also

dhcpserver set subnet maxleasetime on page 131

10.11 dhcpserver set maxleasetime

10.11.1 Syntax

```
dhcpserver set maxleasetime <maxleasetime>
```

10.11.2 Description

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

10.11.3 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
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

10.11.4 Example

```
prompt> dhcpserver set maxleasetime 90000
```

10.11.5 See also

dhcpserver set defaultleasetime on page 126

10.12 dhcpserver set subnet defaultleasetime

10.12.1 Syntax

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

10.12.2 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.

10.12.3 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 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

10.12.4 Example

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

10.12.5 See also

dhcpserver show subnet on page 134

10.13 dhcpserver set subnet hostisdefaultgateway

10.13.1 Syntax

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

10.13.2 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.

10.13.3 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 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.	

10.13.4 Example

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

10.13.5 See also

dhcpserver set subnet hostisdnsserver on page 130

10.14 dhcpserver set subnet hostisdnsserver

10.14.1 Syntax

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

10.14.2 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.

10.14.3 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 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.	

10.14.4 Example

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

10.14.5 See also

dhcpserver list subnets on page 123

10.15 dhcpserver set subnet maxleasetime

10.15.1 Syntax

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

10.15.2 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.

10.15.3 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 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

10.15.4 Example

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

10.15.5 See also

dhcpserver show subnet on page 134

10.16 dhcpserver set subnet subnet

10.16.1 Syntax

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

10.16.2 Description

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

10.16.3 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 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

10.16.4 Example

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

10.16.5 See also

dhcpserver list subnets on page 123

10.17 dhcpserver show

10.17.1 Syntax

```
dhcpserver show
```

10.17.2 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)

10.17.3 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
```

10.17.4 See also

dhcpserver show subnet on page 134

10.18 dhcpserver show subnet

10.18.1 Syntax

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

10.18.2 Description

This command displays the following information about an existing subnet:

- subnet name
- subnet IP address
- subnet netmask
- subnet maximum lease time
- subnet default lease time

10.18.3 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 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

10.18.4 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
```

10.18.5 See also

dhcpserver show on page 133

10.19 dhcpserver subnet add iprange

10.19.1 Syntax

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

10.19.2 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.

10.19.3 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 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
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

10.19.4 Example

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

10.19.5 See also

dhcpserver add subnet on page 115

dhcpserver list subnets on page 123

dhcpserver subnet list ipranges on page 143

10.20 dhcpserver subnet add option

10.20.1 Syntax

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

10.20.2 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* on page 120.

A list of available option data types and value types are also available in *DHCP Server Functional Specification: DO-007343-PS.* 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.

10.20.3 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 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

10.20.4 Example

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

10.20.5 See also

For a list of options that you can choose from, see *dhcpserver list options* on page 120.

For information on RFC 2132, see *http://www.ietf.org/rfc/rfc2132.txt*

DHCP Server Functional Specification: DO-007343-PS

dhcpclient set interfaceconfig autoip enabled|disabled on page 84

10.21 dhcpserver subnet clear ipranges

10.21.1 Syntax

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

10.21.2 Description

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

10.21.3 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 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

10.21.4 Example

```
prompt> dhcpserver subnet sub1 clear ipranges
```

10.21.5 See also

dhcpserver subnet list ipranges on page 143

dhcpserver subnet delete iprange on page 141

10.22 dhcpserver subnet clear options

10.22.1 Syntax

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

10.22.2 Description

This command deletes the options set for an existing subnet.

10.22.3 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 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

10.22.4 Example

```
prompt> dhcpserver subnet sub1 clear options
```

10.22.5 See also

dhcpserver add subnet on page 115

dhcpserver list subnets on page 123

dhcpserver subnet delete option on page 142

10.23 dhcpserver subnet delete iprange

10.23.1 Syntax

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

10.23.2 Description

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

10.23.3 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 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 subnet list ipranges* command.	N/A

10.23.4 Example

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

10.23.5 See also

dhcpserver list subnets on page 123

dhcpserver subnet list ipranges on page 143

10.24 dhcpserver subnet delete option

10.24.1 Syntax

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

10.24.2 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.

10.24.3 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 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

10.24.4 Example

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

10.24.5 See also

dhcpserver add subnet on page 115

dhcpserver clear subnets on page 117

dhcpserver list subnets on page 123

dhcpserver subnet list options on page 144

10.25 dhcpserver subnet list ipranges

10.25.1 Syntax

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

10.25.2 Description

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

10.25.3 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 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

10.25.4 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
-----------------------------------------
```

10.25.5 See also

dhcpserver list subnets on page 123

dhcpserver add subnet on page 115

10.26 dhcpserver subnet list options

10.26.1 Syntax

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

10.26.2 Description

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

10.26.3 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 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

10.26.4 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
-----------------------------------------
```

10.26.5 See also

dhcpserver add subnet on page 115

dhcpserver list subnets on page 123

10.27 dhcpserver update

10.27.1 Syntax

```
dhcpserver update
```

10.27.2 Description

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

10.27.3 Example

```
prompt> dhcpserver update

dhcpserver: Reset request acknowledged. Reset imminent.
```

11.DNS Client CLI commands

This chapter describes the DNS Client CLI commands.

11.1 Summary

11.1.1 DNS Client CLI commands

The table below lists the DNS Client commands provided by the CLI:

Command	Reference
dnsclient add searchdomain	*dnsclient add searchdomain* on page 149
dnsclient add server	*dnsclient add server* on page 150
dnsclient clear searchdomains	*dnsclient clear searchdomains* on page 151
dnsclient clear servers	*dnsclient clear servers* on page 152
dnsclient delete searchdomain	*dnsclient delete searchdomain* on page 153
dnsclient delete server	*dnsclient delete server* on page 154
dnsclient list searchdomains	*dnsclient list searchdomains* on page 155
dnsclient list servers	*dnsclient list servers* on page 156

11.1.2 DNS Client Console commands

The table below lists the *dns client* **console** commands and, if available, their equivalent CLI command:

Command	CLI Equivalent
dnsclient nameserver	Replaced by CLI command *dnsclient add server* on page 150 and *dnsclient delete server* on page 154
dnsclient search	Replaced by CLI command *dnsclient add searchdomain* on page 149
dnsclient show	Replaced by CLI command *dnsclient list searchdomains* on page 155 and *dnsclient list servers* on page 156
dnsclient nslookup	Replaced by CLI command *dnsclient add searchdomain* on page 149 and *dnsclient add server* on page 150
dnsclient cache	Blacklisted command, see *cache* on page 817

11.2 dnsclient add searchdomain

11.2.1 Syntax

```
dnsclient add searchdomain <searchstring>
```

11.2.2 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.

11.2.3 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
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

11.2.4 Example

```
prompt> dnsclient add searchdomain globespanvirata.com
```

11.2.5 See also

dnsclient list searchdomains on page 155

11.3 dnsclient add server

11.3.1 Syntax

```
dnsclient add server <ipaddress>
```

11.3.2 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.

11.3.3 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
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

11.3.4 Example

```
prompt> dnsclient add server 192.168.219.196
```

11.3.5 See also

dnsclient list servers on page 156

11.4 dnsclient clear searchdomains

11.4.1 Syntax

```
dnsclient clear searchdomains
```

11.4.2 Description

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

11.4.3 Example

```
prompt> dnsclient clear searchdomains
```

11.4.4 See also

dnsclient add searchdomain on page 149

dnsclient delete searchdomain on page 153

11.5 dnsclient clear servers

11.5.1 Syntax

```
dnsclient clear servers
```

11.5.2 Description

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

11.5.3 Example

```
prompt> dnsclient clear servers
```

11.5.4 See also

dnsclient add searchdomain on page 149

dnsclient delete server on page 154

11.6 dnsclient delete searchdomain

11.6.1 Syntax

```
dnsclient delete searchdomain <searchstring>
```

11.6.2 Description

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

11.6.3 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
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

11.6.4 Example

```
prompt> dnsclient delete searchdomain 1
```

11.6.5 See also

dnsclient clear searchdomains on page 151

dnsclient list searchdomains on page 155

11.7 dnsclient delete server

11.7.1 Syntax

```
dnsclient delete server <number>
```

11.7.2 Description

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

11.7.3 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
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

11.7.4 Example

```
prompt> dnsclient delete server 1
```

11.7.5 See also

dnsclient clear servers on page 152

dnsclient list servers on page 156

11.8 dnsclient list searchdomains

11.8.1 Syntax

```
dnsclient list searchdomains
```

11.8.2 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.

11.8.3 Example

```
prompt> dnsclient list searchdomains

ID   | Domain
-----|---------------------
   1 | globespanvirata.com
---------------------------
```

11.9 dnsclient list servers

11.9.1 Syntax

```
dnsclient list servers
```

11.9.2 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.

11.9.3 Example

```
prompt> dnsclient list servers


DNS Client Servers:


 ID  | IP Address
----|------------------
  1 | 192.168.100.7
  2 | 192.168.100.1
-------------------------
```

12.DNS Relay CLI commands

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

12.1 Summary

12.1.1 DNS Relay CLI commands

The table below lists the DNS Relay commands provided by the CLI:

Command	Reference
dnsrelay add server	*dnsrelay add server* on page 159
dnsrelay clear servers	*dnsrelay clear servers* on page 160
dnsrelay delete server	*dnsrelay delete server* on page 161
dnsrelay list servers	*dnsrelay list servers* on page 162

12.1.2 DNS relay Console commands

The table below lists the *dns relay* **console** commands and, if available, their equivalent CLI command:

Command	CLI Equivalent
dnsrelay config	Blacklisted command, see *dnsrelay config* on page 820
dnsrelay help	Blacklisted command, see *dnsrelay help* on page 821
dnsrelay pool	Blacklisted command, see *dnsrelay pool* on page 822
dnsrelay retry	Blacklisted command, see *dnsrelay retry* on page 823
dnsrelay server	Replaced by CLI command *dnsrelay add server* on page 159
dnsrelay status	Replaced by CLI command *dnsrelay list servers* on page 162
dnsrelay trace/untrace	Blacklisted command, see *dnsrelay trace/untrace* on page 826
dnsrelay version	Blacklisted command, see *dnsrelay version* on page 828

12.2 dnsrelay add server

12.2.1 Syntax

```
dnsrelay add server <ip-address>
```

12.2.2 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.

12.2.3 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
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

12.2.4 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
```

12.2.5 See also

dnsrelay list servers on page 162

12.3 dnsrelay clear servers

12.3.1 Syntax

```
dnsrelay clear servers
```

12.3.2 Description

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

12.3.3 Example

```
prompt> dnsrelay clear servers
```

12.3.4 See also

dnsrelay delete server on page 161

12.4 dnsrelay delete server

12.4.1 Syntax

```
dnsrelay delete server <id-number>
```

12.4.2 Description

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

12.4.3 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
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

12.4.4 Example

```
prompt> dnsrelay delete server 3
```

12.4.5 See also

dnsrelay list servers on page 162

12.5 dnsrelay list servers

12.5.1 Syntax

```
dnsrelay list servers
```

12.5.2 Description

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

12.5.3 Example

```
prompt> dnsrelay list servers
DNS Relay Servers:


 ID   | IP Address
------|------------------
   1  | 239.252.197.0
------------------------
```

13.Ethernet CLI commands

This chapter describes the Ethernet transport CLI commands.

13.1 Summary

13.1.1 Ethernet CLI commands

The table below lists the CLI commands for manipulating Ethernet channels:

Command	Reference
ethernet add transport	*ethernet add transport* on page 165
ethernet clear transports	*ethernet clear transports* on page 167
ethernet delete transport	*ethernet delete transport* on page 168
ethernet list transports	*ethernet list transports* on page 170
ethernet list ports	*ethernet list ports* on page 169
ethernet set transport	*ethernet set transport port* on page 171
ethernet show transport	*ethernet show transport* on page 172

13.1.2 Ethernet Console commands

The CLI commands above replace **all** previous *ether* console commands.

13.2 ethernet add transport

13.2.1 Syntax

```
ethernet add transport <name> [<port>]
```

13.2.2 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*.

13.2.3 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	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 GlobespanVirata system port that is used to transport ethernet data. You cannot use the same port for more than one ethernet transport at a time.	Ethernet

13.2.4 Example

```
prompt> ethernet add transport eth1 ethernet
```

13.2.5 See also

ethernet list transports on page 170

ethernet list ports on page 169

Console command *set port* on page 764

For information on creating and attaching interfaces and transports, see the Software User's Guide for the GlobespanVirata system that you are using.

For information on the ports available on your GlobespanVirata system, see the Hardware User's Guide for the GlobespanVirata system that you are using.

13.3 ethernet clear transports

13.3.1 Syntax

```
ethernet clear transports
```

13.3.2 Description

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

13.3.3 Example

```
prompt> ethernet clear transports
```

13.3.4 See also

ethernet delete transport on page 168

13.4 ethernet delete transport

13.4.1 Syntax

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

13.4.2 Description

This command deletes a single ethernet transport.

13.4.3 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 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

13.4.4 Example

```
prompt> ethernet delete transport eth1
```

13.4.5 See also

ethernet list transports on page 170

13.5 ethernet list ports

13.5.1 Syntax

```
ethernet list ports
```

13.5.2 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*.

13.5.3 Example

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

13.5.4 See also

Console command *set port* on page 764

For information on the ports available on your GlobespanVirata system, see the Hardware User's Guide for the GlobespanVirata system that you are using.

13.6 ethernet list transports

13.6.1 Syntax

```
ethernet list transports
```

13.6.2 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.

13.6.3 Example

```
prompt> ethernet list transports
Ethernet transports:

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

13.6.4 See also

ethernet list ports on page 169

13.7 ethernet set transport port

13.7.1 Syntax

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

13.7.2 Description

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

13.7.3 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 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 GlobespanVirata system port that is used to transport ethernet data. You cannot use the same port for more than one ethernet transport at a time.	Ethernet

13.7.4 Example

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

13.7.5 See also

ethernet add transport on page 165

ethernet list transports on page 170

For information on the ports available on your GlobespanVirata system, see the Hardware User's Guide for the GlobespanVirata system that you are using.

13.8 ethernet show transport

13.8.1 Syntax

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

13.8.2 Description

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

13.8.3 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
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

13.8.4 Example

```
prompt> ethernet show transport eth1
Ethernet transport: eth1

Description: Default LAN port
      Port: ethernet
```

13.8.5 See also

ethernet list transports on page 170

14.Firewall CLI commands

This chapter describes the stateful Firewall CLI commands.

The Firewall module is a child module in the GlobespanVirata Security package. Before you use the Firewall commands, read *About the VMI Security package on page 542.*

For more information, see the ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS.

14.1 Summary

14.1.1 Firewall CLI commands

The table below lists the *firewall* commands provided by the CLI:

Command	Description/Console command
firewall enable|disable	*firewall enable|disable* on page 176
firewall enable|disable blockinglog	*firewall enable|disable blockinglog* on page 178
firewall enable|disable IDS	*firewall enable|disable IDS* on page 179
firewall enable|disable intrusionlog	*firewall enable|disable on page 176* on page 179
firewall enable|disable sessionlog	*firewall enable|disable sessionlog* on page 181
firewall set securitylevel	*firewall set securitylevel* on page 182
firewall status	*firewall status* on page 187
firewall add policy	*firewall add policy* on page 188
firewall clear policies	*firewall clear policies* on page 190
firewall delete policy	*firewall delete policy* on page 191
firewall list policies	*firewall list policies* on page 192
firewall show policy	*firewall show policy* on page 193
firewall add portfilter	*firewall add portfilter* on page 195
firewall clear portfilters	*firewall clear portfilters* on page 198
firewall delete portfilter	*firewall delete portfilter* on page 199
firewall list portfilters	*firewall list portfilters* on page 200
firewall show portfilter	*firewall show portfilter* on page 202
firewall add validator	*firewall add validator* on page 204
firewall delete validator	*firewall delete validator* on page 207
firewall list validators	*firewall list validators* on page 208
firewall show validator	*firewall show validator* on page 210

Command	Description/Console command
firewall set IDS blacklist	*firewall set IDS blacklist* on page 212
firewall set IDS DOSattackblock	*firewall set IDS DOSattackblock* on page 213
firewall set IDS MaxICMP	*firewall set IDS MaxICMP* on page 214
firewall set IDS MaxPING	*firewall set IDS MaxPING* on page 215
firewall set IDS MaxTCPopenhandshake	*firewall set IDS MaxTCPopenhandshake* on page 216
firewall set IDS SCANattackblock	*firewall set IDS SCANattackblock* on page 218
firewall set IDS victimprotection	*firewall set IDS victimprotection* on page 219
firewall show IDS	*firewall show IDS* on page 221

14.2 firewall enable|disable

14.2.1 Syntax

```
firewall {enable | disable}
```

14.2.2 Description

This command enables/disables the entire Firewall module except for the IDS portion of the module (see the command *firewall enable|disable IDS* on page 179).

Note - You **must** also enable the Security module, using the command *security* on page 545, 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 GlobespanVirata system but want to save the Firewall configuration between sessions, use the *system config save* command.

14.2.3 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
enable	Enables the Firewall module.	disable
disable	Disables the Firewall module.	

14.2.4 Example

```
prompt> firewall enable
```

14.2.5 See also

For information on the Firewall module, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS.*

firewall enable|disable IDS on page 179

firewall set securitylevel on page 182

system config save on page 598

14.3 firewall enable|disable blockinglog

14.3.1 Syntax

```
firewall {enable | disable} blockinglog
```

14.3.2 Description

Note - To display logging information, you need to turn on *event logging* at the console. See *event ...* on page 707.

This command enables/disables whether Firewall blocking activity is logged.

14.3.3 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
enable	The blocking log is displayed.	enable
disable	The blocking log is not displayed.	

14.3.4 Example

```
prompt> firewall enable blockinglog
```

14.3.5 See also

firewall enable|disable sessionlog on page 181

firewall enable|disable on page 176 on page 179

firewall status on page 187

14.4 firewall enable|disable IDS

14.4.1 Syntax

```
firewall {enable | disable} IDS
```

14.4.2 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* on page 545.

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.

14.4.3 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
enable	Enables the IDS portion of the Firewall module.	disable
disable	Disables the IDS portion of the Firewall module.	

14.4.4 Example

```
prompt> firewall enable IDS
```

14.4.5 See also

firewall enable|disable on page 176

14.5 firewall enable|disable intrusionlog

14.5.1 Syntax

```
firewall {enable | disable} intrusionlog
```

14.5.2 Description

Note - To display logging information, you need to turn on *event logging* at the console. See *event ...* on page 707.

This command enables/disables whether details of attempted Firewall intrusion activity are logged.

14.5.3 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
enable	The intrusion log is displayed.	disable
disable	The intrusion log is not displayed.	

14.5.4 Example

```
prompt> firewall enable intrusionlog
```

14.5.5 See also

firewall enable|disable blockinglog on page 178

firewall enable|disable sessionlog on page 181

firewall status on page 187

14.6 firewall enable|disable sessionlog

14.6.1 Syntax

```
firewall {enable | disable} sessionlog
```

14.6.2 Description

Note - To display logging information, you need to turn on *event logging* at the console. See *event ...* on page 707.

This command enables/disables whether Firewall session events are logged.

14.6.3 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
enable	The log containing session details is displayed.	enable
disable	The log containing session details is not displayed.	

14.6.4 Example

```
prompt> firewall enable sessionlog
```

14.6.5 See also

firewall enable|disable blockinglog on page 178

firewall enable|disable on page 176 on page 179

firewall status on page 187

14.7 firewall set securitylevel

14.7.1 Syntax

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

14.7.2 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. For more information on how to configure your own security level, see *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS.*

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.

14.7.3 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:

HIGH SECURITY LEVEL		External < > Internal		External < > DMZ		DMZ < > Internal	
Service	**Port**	**In**	**Out**	**In**	**Out**	**In**	**Out**
http	80	x	3	3	3	3	3
dns	53	x	3	x	3	x	3
ftp	21	x	x	x	3	x	3
telnet	23	x	x	x	x	x	x
smtp	25	x	3	3	3	3	3
pop3	110	x	3	3	3	3	3
nntp	119	x	x	x	x	x	x
real audio/video	7070	x	x	x	x	x	x
icmp	N/A	x	3	x	3	x	3
H.323	1720	x	x	x	x	x	x
T.120	1503	x	x	x	x	x	x
SSH	22	x	x	x	x	x	x

MEDIUM SECURITY LEVEL		External < > Internal		External < > DMZ		DMZ < > Internal	
Service	**Port**	**In**	**Out**	**In**	**Out**	**In**	**Out**
http	80	x	3	3	3	3	3
dns	53	x	3	3	3	3	3
ftp	21	x	3	3	3	3	3

MEDIUM SECURITY LEVEL		External < > Internal		External < > DMZ		DMZ < > Internal	
Service	**Port**	**In**	**Out**	**In**	**Out**	**In**	**Out**
telnet	23	x	3	x	3	x	3
smtp	25	x	3	3	3	3	3
pop3	110	x	3	3	3	3	3
nntp	119	x	3	3	3	3	3
real audio/video	7070	3	x	x	3	x	3
icmp	N/A	x	3	x	3	x	3
H.323	1720	x	3	x	3	x	3
T.120	1503	x	3	x	3	x	3
SSH	22	x	3	x	3	x	3

LOW SECURITY LEVEL		External < > Internal		External < > DMZ		DMZ < > Internal	
Service	**Port**	**In**	**Out**	**In**	**Out**	**In**	**Out**
http	80	x	3	3	3	3	3
dns	53	3	3	3	3	3	3
ftp	21	x	3	3	3	3	3
telnet	23	x	3	3	3	3	3
smtp	25	x	3	3	3	3	3
pop3	110	x	3	3	3	3	3
nntp	119	x	3	3	3	3	3
real audio/video	7070	3	x	3	3	3	3
icmp	N/A	3	3	3	3	3	3
H.323	1720	3	3	3	3	3	3

LOW SECURITY LEVEL		External < > Internal		External < > DMZ		DMZ < > Internal	
Service	**Port**	**In**	**Out**	**In**	**Out**	**In**	**Out**
T.120	1503	3	3	3	3	3	3
SSH	22	3	3	3	3	3	3

The following table gives 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

14.7.4 Example

```
prompt> firewall set securitylevel medium
```

14.7.5 See also

firewall add policy on page 188

For more information on ports assigned to protocols, see *http://www.ietf.org/rfc/rfc1700.txt*

ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS

14.8 firewall status

14.8.1 Syntax

```
firewall status
```

14.8.2 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)

14.8.3 Example

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

14.8.4 See also

firewall enable|disable on page 176

firewall set securitylevel on page 182

firewall enable|disable blockinglog on page 178

firewall enable|disable sessionlog on page 181

firewall enable|disable on page 176 on page 179

14.9 firewall add policy

14.9.1 Syntax

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

14.9.2 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.

14.9.3 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	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
external-internal	A connection between the external network interface and the internal network interface.	N/A
external-dmz	A connection between the external network interface and the de-militarized zone (DMZ).	
dmz-internal	A connection between the de-militarized zone (DMZ) and the internal network interface.	
allowonly-val	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
blockonly-val	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

14.9.4 Example

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

14.9.5 See also

firewall set securitylevel on page 182

firewall add portfilter on page 195

firewall add validator on page 204

14.10 firewall clear policies

14.10.1 Syntax

```
firewall clear policies
```

14.10.2 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.

14.10.3 Example

```
prompt> firewall clear policies
```

14.10.4 See also

firewall add policy on page 188

firewall delete policy on page 191

14.11 firewall delete policy

14.11.1 Syntax

```
firewall delete policy <name>
```

14.11.2 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.

14.11.3 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 firewall policy. To display policy names, use the *firewall list policies* command.	N/A

14.11.4 Example

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

14.11.5 See also

firewall clear policies on page 190

firewall list policies on page 192

14.12 firewall list policies

14.12.1 Syntax

```
firewall list policies
```

14.12.2 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).

14.12.3 Example

```
prompt> firewall list policies
Firewall Policies:

 ID |  Name   | Type 1   | Type 2 | Validator Allow Only
-------------------------------------------------------
  1 | ext-dmz | external | dmz    | true
-------------------------------------------------------
```

14.12.4 See also

firewall show policy on page 193

firewall add policy on page 188

firewall add validator on page 204

14.13 firewall show policy

14.13.1 Syntax

```
firewall show policy <name>
```

14.13.2 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 (blockonly-val is the default setting if no status is specified).

14.13.3 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 firewall policy. To display policy names, use the *firewall list policies* command.	N/A

14.13.4 Example

```
prompt> firewall show policy p2
Firewall Policy: ext-dmz

Interface Type 1: external
Interface Type 2: dmz

Allow Only Validator: true
```

14.13.5 See also

firewall list policies on page 192

firewall add validator on page 204

14.14 firewall add portfilter

14.14.1 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}
```

14.14.2 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.

14.14.3 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	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

Option	Description	Default value
(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

14.14.4 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
```

14.14.5 See also

firewall list policies on page 192

See the *Well Known Port Numbers* section of RFC 1700 for a list of port numbers and protocols for particular services (see *http://www.ietf.org/rfc/rfc1700.txt*).

14.15 firewall clear portfilters

14.15.1 Syntax

```
firewall clear portfilters <policyname>
```

14.15.2 Description

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

14.15.3 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
policyname	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A

14.15.4 Example

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

14.15.5 See also

firewall delete portfilter on page 199

firewall list policies on page 192

14.16 firewall delete portfilter

14.16.1 Syntax

```
firewall delete portfilter <name> <policyname>
```

14.16.2 Description

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

14.16.3 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 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

14.16.4 Example

```
prompt> firewall delete portfilter pf3 ext-int
```

14.16.5 See also

firewall list policies on page 192

firewall list portfilters on page 200

firewall clear portfilters on page 198

14.17 firewall list portfilters

14.17.1 Syntax

```
firewall list portfilters <policyname>
```

14.17.2 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)

14.17.3 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
policyname	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A

14.17.4 Example

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

```
Firewall Port Filters:

  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
-------------------------------------------------------------------------------
```

14.17.5 See also

firewall list policies on page 192

firewall show portfilter on page 202

For a list of the port numbers and/or numbers assigned to protocols, see *http://www.ietf.org/rfc/rfc1700.txt*.

14.18 firewall show portfilter

14.18.1 Syntax

```
firewall show portfilter <name> <policyname>
```

14.18.2 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)

14.18.3 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 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

14.18.4 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
```

14.18.5 See also

firewall list policies on page 192

firewall list portfilters on page 200

14.19 firewall add validator

14.19.1 Syntax

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

14.19.2 Description

Note - Before you can add validators to the Firewall, 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**.

See *firewall add policy* on page 188.

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.

14.19.3 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	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

14.19.4 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
```

14.19.5 See also

firewall add policy on page 188

firewall list policies on page 192

firewall delete validator on page 207

firewall show validator on page 210

14.20 firewall delete validator

14.20.1 Syntax

```
firewall delete validator <name> <policyname>
```

14.20.2 Description

This command deletes a single validator from a named policy.

14.20.3 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 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

14.20.4 Example

```
prompt> firewall delete validator v1 ext-int
```

14.20.5 See also

firewall list validators on page 208

firewall list policies on page 192

14.21 firewall list validators

14.21.1 Syntax

```
firewall list validators <policyname>
```

14.21.2 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

14.21.3 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
policyname	A name that identifies an existing firewall policy. To display policy names, use the *firewall list policies* command.	N/A

14.21.4 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
-------------------------------------------------------------
```

14.21.5 See also

firewall add validator on page 204

firewall show validator on page 210

firewall list policies on page 192

14.22 firewall show validator

14.22.1 Syntax

```
firewall show validator <name> <policyname>
```

14.22.2 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

14.22.3 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 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

14.22.4 Example

```
prompt> firewall show validator v1


Firewall Host Validator: v1


Direction: both
  Host IP: 192.168.103.2
Host Mask: 255.255.255.0
```

14.22.5 See also

firewall add validator on page 204

firewall list validators on page 208

firewall list policies on page 192

14.23 firewall set IDS blacklist

14.23.1 Syntax

```
firewall set IDS blacklist {enable | disable | clear}
```

14.23.2 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.

14.23.3 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
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.	

14.23.4 Example

```
prompt> firewall set IDS blacklist enable
```

14.23.5 See also

For more information on IDS, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

14.24 firewall set IDS DOSattackblock

14.24.1 Syntax

```
firewall set IDS DOSattackblock <duration>
```

14.24.2 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.

14.24.3 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
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)

14.24.4 Example

```
prompt> firewall set DOSattackblock 3600
```

14.24.5 See also

For more information on IDS, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

14.25 firewall set IDS MaxICMP

14.25.1 Syntax

```
firewall set IDS MaxICMP <max>
```

14.25.2 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.

14.25.3 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
max	The maximum number (per second) of ICMP packets that are allowed before an ICMP Flood attempt is detected.	100

14.25.4 Example

```
prompt> firewall set IDS MaxICMP 200
```

14.25.5 See also

For more information on IDS, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

14.26 firewall set IDS MaxPING

14.26.1 Syntax

```
firewall set IDS MaxPING <max>
```

14.26.2 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.

14.26.3 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
max	The maximum number (per second) of pings that are allowed before an Echo Storm attempt is detected.	15

14.26.4 Example

```
prompt> firewall set IDS MaxPING 25
```

14.26.5 See also

For more information on IDS, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

14.27 firewall set IDS MaxTCPopenhandshake

14.27.1 Syntax

```
firewall set IDS MaxTCPopenhandshake <max>
```

14.27.2 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.

14.27.3 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
max	The maximum number (per second) of unfinished TCP handshaking sessions that are allowed before a SYN Flood attempt is detected.	100

14.27.4 Example

```
prompt> firewall set IDS MaxTCPopenhandshake 150
```

14.27.5 See also

For more information on IDS, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

14.28 firewall set IDS SCANattackblock

14.28.1 Syntax

```
firewall set IDS SCANattackblock <duration>
```

14.28.2 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.

14.28.3 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
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)

14.28.4 Example

```
prompt> firewall set IDS SCANattackblock 43200
```

14.28.5 See also

For more information on IDS, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

14.29 firewall set IDS victimprotection

14.29.1 Syntax

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

14.29.2 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.

14.29.3 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
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)

14.29.4 Example

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

14.29.5 See also

For more information on IDS, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

14.30 firewall show IDS

14.30.1 Syntax

```
firewall show IDS
```

14.30.2 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)

14.30.3 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
```

15.Frame Relay CLI commands

This chapter describes the Frame Relay CLI commands.

15.1 Summary

15.1.1 Frame Relay CLI commands

The table below lists the *Frame Relay* commands provided by the CLI:

Command	Description/Console command
framerelay add transport	*framerelay add transport* on page 225
framerelay clear transports	*framerelay clear transports* on page 226
framerelay delete transport	*framerelay delete transport* on page 227
framerelay list transports	*framerelay list transports* on page 228
framerelay set transport chnlsegmentsize	*framerelay set transport chnlsegmentsize* on page 229
framerelay set transport dlci	*framerelay set transport dlci* on page 230
framerelay set transport encapsulation	*framerelay set transport encapsulation* on page 231
framerelay set transport port	*framerelay set transport port* on page 233
framerelay set transport rxmaxpdu	*framerelay set transport rxmaxpdu* on page 234
framerelay set transport tcmaxpdu	*framerelay set transport tcmaxpdu* on page 235
framerelay show transport	*framerelay show transport* on page 236

15.2 framerelay add transport

15.2.1 Syntax

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

15.2.2 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.

15.2.3 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	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 GlobespanVirata system port that 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

15.2.4 Example

```
prompt> framerelay add transport fr1 fr_relay 171
```

15.2.5 See also

framerelay list transports on page 228

framerelay show transport on page 236.

15.3 framerelay clear transports

15.3.1 Syntax

```
framerelay clear transports
```

15.3.2 Description

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

15.3.3 Example

```
prompt> framrelay clear transports
```

15.3.4 See also

framerelay delete transport on page 227

15.4 framerelay delete transport

15.4.1 Syntax

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

15.4.2 Description

This command deletes a single Frame Relay transport.

15.4.3 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 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

15.4.4 Example

```
prompt> framerelay delete transport fr1
```

15.4.5 See also

framerelay list transports on page 228

15.5 framerelay list transports

15.5.1 Syntax

```
framerelay list transports
```

15.5.2 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.

15.5.3 Example

```
prompt> framerelay list transports
Frame Relay Transports:

 ID  |    Name    |    Port    |  DLCI  | Encapsulation
-----|------------|------------|--------|-----------------
   1 | fr1        | fr_relay   | 171    | Raw
-------------------------------------------------------------
```

15.5.4 See also

framerelay list transports on page 228

15.6 framerelay set transport chnlsegmentsize

15.6.1 Syntax

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

15.6.2 Description

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

15.6.3 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 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

15.6.4 Example

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

15.6.5 See also

framerelay list transports on page 228

For more information on FRF.12, *see http://www.frforum.com*.

15.7 framerelay set transport dlci

15.7.1 Syntax

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

15.7.2 Description

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

15.7.3 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 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

15.7.4 Example

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

15.7.5 See also

framerelay list transports on page 228

15.8 framerelay set transport encapsulation

15.8.1 Syntax

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

15.8.2 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.

15.8.3 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
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.	

15.8.4 Example

```
prompt> framerelay set transport encapsulation bridgedether
```

15.8.5 See also

framerelay list transports on page 228

For information on how to attach Frame Relay to the bridge or to the IP router, see *BUN Devices: Frame Relay: DO-008218-PS*.

For details of RFC1490, see http://www.ietf.org/rfc/rfc1490.txt.

15.9 framerelay set transport port

15.9.1 Syntax

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

15.9.2 Description

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

15.9.3 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 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 GlobespanVirata system port that is used to transport Frame Relay data.	N/A

15.9.4 Example

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

15.9.5 See also

framerelay list transports on page 228

15.10 framerelay set transport rxmaxpdu

15.10.1 Syntax

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

15.10.2 Description

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

15.10.3 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 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

15.10.4 Example

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

15.10.5 See also

framerelay list transports on page 228

15.11 framerelay set transport tcmaxpdu

15.11.1 Syntax

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

15.11.2 Description

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

15.11.3 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 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

15.11.4 Example

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

15.11.5 See also

framerelay list transports on page 228

15.12 framerelay show transport

15.12.1 Syntax

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

15.12.2 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

15.12.3 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 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

15.12.4 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
```

15.12.5 See also

.*framerelay list transports* on page 228

16.IGMP CLI commands

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

16.1 Summary

16.1.1 IGMP CLI commands

The table below lists the *IGMP* commands provided by the CLI:

Command	Description/Console command
igmp set upstreaminterface	*igmp set upstreaminterface* on page 241
igmp show upstreaminterface	*igmp show upstreaminterface* on page 242
igmp show status	*igmp show status* on page 243

16.2 igmp set upstreaminterface

16.2.1 Syntax

```
igmp set upstreaminterface {<ip_interface> | none}
```

16.2.2 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*.

16.2.3 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
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

16.2.4 Example

```
prompt> igmp set upstreaminterface ip1
```

16.2.5 See also

IP Stack Programmer's Guide: DO-010017-TC

16.3 igmp show upstreaminterface

16.3.1 Syntax

```
igmp show upstreaminterface
```

16.3.2 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.

16.3.3 Example

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

16.3.4 See also

igmp set upstreaminterface on page 241

IP Stack Programmer's Guide: DO-010017-TC

16.4 igmp show status

16.4.1 Syntax

```
igmp show status
```

16.4.2 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

16.4.3 Example

```
prompt> igmp show status
Multicast group membership:

 Interface (querier) | Group address
---------------------|-----------------
 eth0          (yes) | 239.255.255.250
 r1483         (yes) | 224.0.1.101
 r1483         (yes) | 224.0.1.102
 r1483         (yes) | 224.0.1.103
--------------------------------------
```

16.4.4 See also

igmp show upstreaminterface on page 242

IP Stack Programmer's Guide: DO-010017-TC

17.IPoA CLI commands

This chapter describes the IP over ATM (IPoA) transport CLI commands

17.1 Summary

17.1.1 IPoA Tracing commands

You can carry out tracing in the IP stack using the following *system* commands:

- *system log enable|disable* on page 601; enables/disables tracing support output for a specific module and category, including IPoA.
- *system log list* on page 603; displays the tracing options for the modules available in your image.

17.1.2 IPoA CLI commands

The table below lists the IPoA commands provided by the CLI:

Command	Reference
ipoa add transport pvc	*ipoa add transport pvc* on page 248
ipoa add transport svc	*ipoa add transport svc* on page 250
ipoa clear transports	*ipoa clear transports* on page 251
ipoa delete transport	*ipoa delete transport* on page 252
ipoa list arp	*ipoa list arp* on page 253
ipoa list transports	*ipoa list transports* on page 254
ipoa set transport arpserver	*ipoa set transport arpserver* on page 255
ipoa set transport svc	*ipoa set transport svc* on page 257
ipoa set transport lifetime	*ipoa set transport lifetime* on page 256
ipoa show transport	*ipoa show transport* on page 259
ipoa transport add pvc	*ipoa transport add pvc* on page 261
ipoa transport delete pvc	*ipoa transport delete pvc* on page 263

Command	Reference
ipoa transport set pvc pcr	*ipoa transport set pvc pcr* on page 264
ipoa transport set pvc port	*ipoa transport set pvc port* on page 266
ipoa transport set pvc vpi	*ipoa transport set pvc vpi* on page 268
ipoa transport set pvc vci	*ipoa transport set pvc vci* on page 269

17.2 ipoa add transport pvc

17.2.1 Syntax

```
ipoa add transport <name> pvc <atm-port> <vpi> <vci>
```

17.2.2 Description

This command creates a named IPoA transport that runs over a PVC (Permanent Virtual Circuit). It allows you to specify the following PVC information:

- the ATM port that will transport IP data
- VPI (Virtual Path Identifier)
- VCI (Virtual Circuit Identifier)

The port/VPI/VCI combination must be unique for each transport.

17.2.3 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	An arbitrary name that identifies an IPoA transport. It can be made up of one or more characters or a combination of characters and digits, but it cannot start with a digit.	N/A
pvc	A connection-oriented permanent leased line circuit between end stations on a network. An IPoA transport can have multiple PVCs associated with it.	N/A
port	The Virata system port that is used to transport ATM data.	N/A
vpi	A field in the ATM header. The VPI is used to identify the virtual path that a circuit belongs to. The VPI can be any value between 0 and 4095.	0
vci	Part of the ATM header. The VCI is a tag that identifies which channel a cell will travel over.The VCI can be any value between 1 and 65535.	0

17.2.4 Example

```
prompt> ipoa add transport ipoa1 pvc a1 0 800
```

17.2.5 See also

ipoa list transports on page 254

To list ATM ports, see the usable console command BUN *list ports* on page 752.

17.3 ipoa add transport svc

17.3.1 Syntax

```
ipoa add transport <name> svc
```

17.3.2 Description

This command creates a named IPoA transport that runs over an SVC (Switched Virtual Circuit). A switched circuit is set up between end stations on the network dial-up. An IPoA transport can only have one SVC associated with it.

Note - You **must** have the *package_si* module included in your image build if you want to create SVC transports.

17.3.3 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	An arbitrary name that identifies an IPoA transport. It can be made up of one or more characters or a combination of characters and digits, but it cannot start with a digit.	N/A

17.3.4 Example

```
prompt> ipoa add transport ipoa2 svc
```

17.3.5 See also

ipoa list transports on page 254

For information on adding modules to ISOS build images, see the Software User's Guide for the GlobespanVirata system that you are using.

17.4 ipoa clear transports

17.4.1 Syntax

```
ipoa clear transports
```

17.4.2 Description

This command deletes all IPoA transports that were created using the *ipoa add transport* commands.

17.4.3 Example

```
prompt> ipoa clear transports
```

17.4.4 See also

ipoa delete transport on page 252

17.5 ipoa delete transport

17.5.1 Syntax

```
ipoa delete transport {<name>|<number>}
```

17.5.2 Description

This command deletes a single IPoA transport.

17.5.3 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 IPoA transport. To display transport names, use the *ipoa list transports* command.	N/A
number	A number that identifies an existing IPoA transport. To display transport numbers, use the *ipoa list transports* command.	N/A

17.5.4 Example

```
prompt> ipoa delete transport ipoa1
```

17.5.5 See also

ipoa list transports on page 254

17.6 ipoa list arp

17.6.1 Syntax

```
ipoa list arp
```

17.6.2 Description

This command lists the cached mappings from IP addresses to ATM addresses that were previously set using the *ipoa set transport arpserver* command. This command is only relevant when using IP over ATM with SVCs.

Note - You **must** have the *package_si* module included in your image build if you want to use SVC transports.

17.6.3 Example 1

If there are entries in the ATM ARP table, the output from this command looks like this:

```
prompt> ipoa list arp
10.11.12.13 47.00.83.00.00.00.00.00.20.2b.00.04.93.00.20.2b.00.04.93.00
```

17.6.4 Example 2

If there are no entries in the ATM ARP table, the output from this command looks like this:

```
prompt> ipoa list arp
# ATMARP table is empty
```

17.6.5 See also

ipoa set transport arpserver on page 255

17.7 ipoa list transports

17.7.1 Syntax

```
ipoa list transports
```

17.7.2 Description

This command lists IPoA transports that have been created using the *ipoa add transport* commands. It displays the following information about the transports:

- transport identification number
- transport name
- SVC status (enabled or disabled)
- ATM Address Resolution Protocol (ARP) Server information (if set using the *ipoa set transport arpserver* command).

17.7.3 Example

```
prompt> ipoa list transports

IPoA transports:

 ID  |    Name    |   SVC    |                 ATM ARP Server
-----|------------|----------|-------------------------------------------
   1 | ipoa1      | disabled | here
 ------------------------------------------------------------------------
```

17.7.4 See also

To display more information about an individual transport, use the command *ipoa show transport* on page 259

ipoa set transport arpserver on page 255

17.8 ipoa set transport arpserver

17.8.1 Syntax

```
ipoa set transport {<name>|<number>} arpserver <server>
```

17.8.2 Description

This command sets the Address Resolution Protocol (ARP) Server that an IPoA transport will use. It only applies to SVCs.

Note - You **must** have the *package_si* module included in your image build in order to use SVC transports.

The IPoA transport needs to be attached to an IP interface in order to transport data. See *TCP/IP CLI commands* on page 613.

17.8.3 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
name	A name that identifies an existing IPoA transport. To display transport names, use the *ipoa list transports* command.	N/A
number	A number that identifies an existing IPoA transport. To display transport numbers, use the *ipoa list transports* command.	N/A
server	The ATM ARP server you want to use. If you want your local machine to be the ATM ARP server, type *'here'*. If you want another machine to be the ATM ARP server, type the ATM-address of that machine.	N/A

17.8.4 Example

```
prompt> ipoa set transport ipoa1 arpserver
47.00.83.10.a2.b1.00.00.00.00.00.00.00.00.20.2b.01.00.07.00
```

17.8.5 See also

ipoa list transports on page 254

ipoa list arp on page 253

17.9 ipoa set transport lifetime

17.9.1 Syntax

ipoa set transport {<name>|<number>} lifetime <timeout>

17.9.2 Description

This command sets idle time-out for IP over ATM SVCs. If there is no traffic on an SVC for this period, then it will be disconnected. (It might be disconnected before this period in order to make room for new connections.)

Note - You **must** have the *package_si* module included in your image build in order to use SVC transports.

17.9.3 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 IPoA transport. To display transport names, use the *ipoa list transports* command.	N/A
number	A number that identifies an existing IPoA transport. To display transport numbers, use the *ipoa list transports* command.	N/A
timeout	The idle time-out (in seconds) for IP over ATM SVC connections.	60

17.9.4 Example

```
prompt> ipoa set transport ipoa1 lifetime 120
```

17.9.5 See also

ipoa show transport on page 259

17.10 ipoa set transport svc

17.10.1 Syntax

```
ipoa set transport {<name>|<number>} svc {<enabled>|<disabled>}
```

17.10.2 Description

This command sets an existing IPoA transport that runs over PVC(s) to run over an SVC (Switched Virtual Circuit). The PVC(s) settings are reserved so that you can switch the transport back to run over PVC(s) if required.

Note - You **must** have the *package_si* module included in your image build in order to use SVC transports.

17.10.3 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 IPoA transport. To display transport names, use the *ipoa list transports* command.	N/A
number	A number that identifies an existing IPoA transport. To display transport numbers, use the *ipoa list transports* command.	N/A
svc	A switched circuit set up between end stations on the network dial-up. An IPoA transport can only have one SVC associated with it.	N/A
enabled	Sets an existing IPoA transport that runs over PVC(s) to run over an SVC.	disabled
disabled	Disables the SVC and switches a transport back to run over PVC(s).	

17.10.4 Example

```
prompt> ipoa set transport ipoa1 svc enabled
```

17.10.5 See also

ipoa list transports on page 254

For information on adding modules to ISOS build images, see the Software User's Guide for the GlobespanVirata system that you are using.

17.11 ipoa show transport

17.11.1 Syntax

```
ipoa show transport {<name>|<number>}
```

17.11.2 Description

This command displays the following information about an existing IPoA transport:

- transport name
- transport description
- Switched Virtual Channel (SVC) status (true or false)
- ARP Server ATM details (Server IP address or 'here')
- details about Permanent Virtual Circuits (PVC) attached to this transport:
  - PVC ID number
  - Port used by PVC
  - Virtual Path Identifier (VPI) setting
  - Virtual Circuit Identifier (VCI) setting
  - Peak Cell Rate (PCR) setting

17.11.3 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 IPoA transport. To display transport names, use the *ipoa list transports* command.	N/A
number	A number that identifies an existing IPoA transport. To display transport numbers, use the *ipoa list transports* command.	N/A

17.11.4 Example

```
prompt> ipoa show transport ipoa1
IPoA Transport: ipoa1
```

```
      Description: ipoa1
              SVC: false
   Arp server ATM: here
Idle SVC lifetime: 60 secs

PVCs :

 ID  |  Port  |  Vpi   |  Vci   |  PCR
-----|--------|--------|--------|--------
   1 | a1     | 0      | 900    | 59111
   2 | a1     | 0      | 600    | 59111
   3 | a1     | 0      | 800    | 59111
----------------------------------------
```

If the IPoA transport has SVC enabled but also has PVC(s) reserved for future use, this command also displays the following information:

```
SVC is enabled - the PVC's listed below are NOT in use
```

17.11.5 See also

ipoa list transports on page 254

17.12 ipoa transport add pvc

17.12.1 Syntax

```
ipoa transport {<name>|<number>} add pvc <port> <vpi> <vci>
```

17.12.2 Description

This command adds a new Permanent Virtual Circuit (PVC) to an existing IPoA transport. You can add new PVCs to existing PVC and SVC IPoA transports. On SVC transports, the PVC(s) are reserved so that you can disable the SVC (using the *ipoa set transport svc* command) and run over PVC(s) if required.

PVC is a connection-oriented permanent leased line circuit between end stations on a network. This command allows you to specify the following PVC information:

- the ATM port that will transport IP data
- VPI (Virtual Path Identifier)
- VCI (Virtual Circuit Identifier)

The port/VPI/VCI combination must be unique for each transport.

17.12.3 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 IPoA transport. To display transport names, use the *ipoa list transports* command.	N/A
number	A number that identifies an existing IPoA transport. To display transport numbers, use the *ipoa list transports* command.	N/A
port	The Virata system port that is used to transport ATM data.	N/A

Option	Description	Default value
vpi	A field in the ATM header. The VPI is used to identify the virtual path that a circuit belongs to. The VPI can be any value between 0 and 4095.	0
vci	Part of the ATM header. The VCI is a tag that identifies which channel a cell will travel over.The VCI can be any value between 1 and 65535.	0

17.12.4 Example

```
prompt> ipoa transport ipoa1 add pvc a1 0 800
```

17.12.5 See also

ipoa list transports on page 254

ipoa show transport on page 259

ipoa set transport svc on page 257

To list ATM ports, see the usable console command *list ports* on page 752.

17.13 ipoa transport delete pvc

17.13.1 Syntax

```
ipoa transport {<name>|<number>} delete pvc <pvc number>
```

17.13.2 Description

This command deletes a PVC from an IPoA transport that was added using the *ipoa transport add pvc* command.

17.13.3 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 IPoA transport. To display transport names, use the *ipoa list transports* command.	N/A
number	A number that identifies an existing IPoA transport. To display transport numbers, use the *ipoa list transports* command.	N/A
pvc number	A number that identifies an existing PVC. The PVC numbers associated with an IPoA transport are displayed using the *ipoa show transport* command. PVC numbers appear in the *ID* column of the PVC table. For an example of this table, see *ipoa show transport* on page 259.	N/A

17.13.4 Example

```
prompt> ipoa transport ipoa1 delete pvc 2
```

17.13.5 See also

ipoa list transports on page 254

ipoa show transport on page 259

ipoa transport add pvc on page 261

17.14 ipoa transport set pvc pcr

17.14.1 Syntax

```
ipoa transport {<name>|<number>} set pvc <pvc number> pcr <peak cell rate>
```

17.14.2 Description

This command sets the peak cell rate for an existing PVC.

17.14.3 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 IPoA transport. To display transport names, use the *ipoa list transports* command.	N/A
number	A number that identifies an existing IPoA transport. To display transport numbers, use the *ipoa list transports* command.	N/A
pvc number	A number that identifies an existing PVC. The PVC numbers associated with an IPoA transport are displayed using the *ipoa show transport* command. PVC numbers appear in the *ID* column of the PVC table. For an example of this table, see *ipoa show transport* on page 259.	N/A
peak cell rate	Determines the maximum rate at which ATM cells are allowed to be transported into the ATM network. The PCR can be any value from 0 up to the maximum PortSpeed parameter set when the port was created (using the initbun file in FlashFS or the console command BUN *set port* on page 754)	0

17.14.4 Example

```
prompt> ipoa transport ipoa1 set pvc 1 pcr 50000
```

17.14.5 See also

ipoa list transports on page 254

ipoa show transport on page 259

17.15 ipoa transport set pvc port

17.15.1 Syntax

```
ipoa transport {<name>|<number>} set pvc <pvc number> port <port>
```

17.15.2 Description

This command changes the port that an existing PVC uses to transport IPoA data.

17.15.3 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 IPoA transport. To display transport names, use the *ipoa list transports* command.	N/A
number	A number that identifies an existing IPoA transport. To display transport numbers, use the *ipoa list transports* command.	N/A
pvc number	A number that identifies an existing PVC. The PVC numbers associated with an IPoA transport are displayed using the *ipoa show transport* command. PVC numbers appear in the *ID* column of the PVC table. For an example of this table, see *ipoa show transport* on page 259.	N/A
port	The Virata system port that is used to transport ATM data. The port/VPI/VCI combination must be unique for each transport.	N/A

17.15.4 Example

```
prompt> ipoa transport ipoa1 set pvc 1 port a1
```

17.15.5 See also

ipoa list transports on page 254

ipoa show transport on page 259

To list ATM ports, see the usable console command BUN *list ports* on page 752.

17.16 ipoa transport set pvc vpi

17.16.1 Syntax

```
ipoa transport {<name>|<number>} set pvc <pvc number> vpi <vpi>
```

17.16.2 Description

This command sets the VPI (Virtual Path Identifier) for an existing PVC. The port/VPI/VCI combination must be unique for each transport.

17.16.3 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 IPoA transport. To display transport names, use the *ipoa list transports* command.	N/A
number	A number that identifies an existing IPoA transport. To display transport numbers, use the *ipoa list transports* command.	N/A
pvc number	A number that identifies an existing PVC. The PVC numbers associated with an IPoA transport are displayed using the *ipoa show transport* command. PVC numbers appear in the *ID* column of the PVC table. For an example of this table, see *ipoa show transport* on page 259.	N/A
vpi	A field in the ATM header. The VPI is used to identify the virtual path that a circuit belongs to. The VPI can be any value between 0 and 4095.	0

17.16.4 Example

```
prompt> ipoa transport ipoa1 set pvc 1 vpi 0
```

17.16.5 See also

ipoa list transports on page 254

ipoa show transport on page 259

17.17 ipoa transport set pvc vci

17.17.1 Syntax

```
ipoa transport {<name>|<number>} set pvc <pvc number> vci <vci>
```

17.17.2 Description

This command sets the VCI (Virtual Circuit Identifier) for an existing PVC.

17.17.3 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 IPoA transport. To display transport names, use the *ipoa list transports* command.	N/A
number	A number that identifies an existing IPoA transport. To display transport numbers, use the *ipoa list transports* command.	N/A
pvc number	A number that identifies an existing PVC. The PVC numbers associated with an IPoA transport are displayed using the *ipoa show transport* command. PVC numbers appear in the *ID* column of the PVC table. For an example of this table, see *ipoa show transport* on page 259.	N/A
vci	Part of the ATM header. The VCI is a tag that identifies which channel a cell will travel over.The VCI can be any value between 1 and 65535.	0

17.17.4 Example

```
prompt> ipoa transport ipoa1 set pvc 1 vci 800
```

17.17.5 See also

ipoa list transports on page 254

ipoa show transport on page 259

18.MAC Spoofing CLI commands

This chapter describes CLI support for MAC spoofing. MAC spoofing is only available to ISOS Switch Router (ISR) users.

18.1 Summary

18.1.1 MAC Spoof CLI commands

The table below lists the *macspoof* commands provided by the CLI:

Command	Reference
macspoof show	*macspoof show* on page 273
macspoof set	*macspoof set* on page 274

18.2 macspoof show

18.2.1 Syntax

```
macspoof show
```

18.2.2 Description

This command displays the MAC address currently configured for your system.

18.2.3 Example

```
prompt> macspoof show
MacSpoof:
   MAC: 00:25:2b:80:33:a0
```

18.2.4 See also

macspoof set on page 274

18.3 macspoof set

18.3.1 Syntax

```
macspoof set <attrname> <newvalue>
```

18.3.2 Description

This command allows you to change the MAC address of your system.

18.3.3 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
attrname	Type *MAC* as the attribute name that you want to set.	N/A
newvalue	The new MAC address that you want to set.	N/A

18.3.4 Example

```
prompt> macspoof set MAC 00:25:2b:80:33:a0
```

18.3.5 See also

macspoof show on page 273

19.NAT CLI commands

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

The NAT module is a child module in the GlobespanVirata Security package. Before you use the NAT commands, read *About the VMI Security package on page 542.*

For more information, see the ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS.

19.1 Summary

19.1.1 NAT CLI commands

The table below lists the NAT commands provided by the CLI:

Command	Reference
nat enable	*nat enable* on page 277
nat disable	*nat disable* on page 279
nat add globalpool	*nat add globalpool* on page 280
nat clear globalpools	*nat clear globalpools* on page 283
nat delete globalpool	*nat delete globalpool* on page 284
nat list globalpools	*nat list globalpools* on page 285
nat show globalpool	*nat show globalpool* on page 287
nat add resvmap globalip	*nat add resvmap globalip* on page 289
nat add resvmap interfacename	*nat add resvmap interfacename* on page 292
nat clear resvmaps	*nat clear resvmaps* on page 295
nat delete resvmap	*nat delete resvmap* on page 296
nat list resvmaps	*nat list resvmaps* on page 297
nat show resvmap	*nat show resvmap* on page 299
nat status	*nat status* on page 301

NAT Console commands and pre-8.1 CLI commands

The IP stack used in ISOS release 8.1 and later **does not** support any of the NAT console commands and CLI commands that were supported in previous releases. All NAT console commands and CLI commands available before ISOS release 8.1 are obsolete.

19.2 nat enable

19.2.1 Syntax

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

19.2.2 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.

Note - You **must** enable the Security package using the command *security* on page 545 if you want to use the NAT module to configure security for your system.

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* on page 292.

19.2.3 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	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 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) 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

19.2.4 Example

```
prompt> nat enable nat1 extinterface internal
```

19.2.5 See also

nat disable on page 279

nat status on page 301

security list interfaces on page 550

security add interface on page 547

nat add resvmap interfacename on page 292

19.3 nat disable

19.3.1 Syntax

```
nat disable <name>
```

19.3.2 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.

19.3.3 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	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

19.3.4 Example

```
prompt> nat disable nat1
```

19.3.5 See also

nat enable on page 277

nat status on page 301

19.4 nat add globalpool

19.4.1 Syntax

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

19.4.2 Description

Note - Before you can add a global address pool, you must enable a NAT object using the command *nat enable* on page 277.

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* on page 289.

19.4.3 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	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

19.4.4 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
```

19.4.5 See also

nat enable on page 277

nat status on page 301

security list interfaces on page 550

Once you have created an address pool, packets received on a specific IP address can be mapped to individual hosts inside the network. See *nat add resvmap globalip* on page 289.

19.5 nat clear globalpools

19.5.1 Syntax

```
nat clear globalpools <interfacename>
```

19.5.2 Description

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

19.5.3 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

19.5.4 Example

```
prompt> nat clear globalpools extinterface
```

19.5.5 See also

nat add globalpool on page 280

security list interfaces on page 550

19.6 nat delete globalpool

19.6.1 Syntax

```
nat delete globalpool <name> <interfacename>
```

19.6.2 Description

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

19.6.3 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 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

19.6.4 Example

```
prompt> nat delete globalpool gp1 extinterface
```

19.6.5 See also

nat add globalpool on page 280

nat list globalpools on page 285

security list interfaces on page 550

19.7 nat list globalpools

19.7.1 Syntax

```
nat list globalpools <interfacename>
```

19.7.2 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

19.7.3 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

19.7.4 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
-------------------------------------------------------------------------------
```

19.7.5 See also

security list interfaces on page 550

nat show globalpool on page 287

19.8 nat show globalpool

19.8.1 Syntax

```
nat show globalpool <name> <interfacename>
```

19.8.2 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

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
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

19.8.4 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
```

19.8.5 See also

nat list globalpools on page 285

security list interfaces on page 550

19.9 nat add resvmap globalip

19.9.1 Syntax

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

19.9.2 Description

Note - Before you can add reserved mapping, you must enable a NAT object using the command *nat enable* on page 277.

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.

19.9.3 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	An arbitrary name that identifies a reserved mapping configuration. 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
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

Option	Description	Default value
(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 autonomous systems. 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

Option	Description	Default value
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

19.9.4 Example

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

19.9.5 See also

nat enable on page 277

nat list globalpools on page 285

nat status on page 301

security list interfaces on page 544

19.10 nat add resvmap interfacename

19.10.1 Syntax

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

19.10.2 Description

Note - Before you can add reserved mapping, you must enable a NAT object using the command *nat enable* on page 277.

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.

19.10.3 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	An arbitrary name that identifies a reserved mapping configuration. 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
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

Option	Description	Default value
(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 autonomous systems. 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

19.10.4 Example

```
prompt> nat add resvmap rm1 interfacename extinterface
10.10.10.10 tcp 25
```

19.10.5 See also

nat enable on page 277

nat status on page 301

security list interfaces on page 544

19.11 nat clear resvmaps

19.11.1 Syntax

```
nat clear resvmaps <interfacename>
```

19.11.2 Description

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

19.11.3 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

19.11.4 Example

```
prompt> nat clear resvmaps extinterface
```

19.11.5 See also

nat delete resvmap on page 296

security list interfaces on page 550

19.12 nat delete resvmap

19.12.1 Syntax

```
nat delete resvmap <name> <interfacename>
```

19.12.2 Description

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

19.12.3 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 global IP address. To display global IP addresses, use the *nat list resvmaps* 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

19.12.4 Example

```
prompt> nat delete resvmap rm1 extinterface
```

19.12.5 See also

nat enable on page 277

nat list resvmaps on page 297

security list interfaces on page 550

19.13 nat list resvmaps

19.13.1 Syntax

```
nat list resvmaps <interfacename>
```

19.13.2 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*.

19.13.3 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

19.13.4 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
-------------------------------------------------------------------------
```

19.13.5 See also

security list interfaces on page 550

19.14 nat show resvmap

19.14.1 Syntax

```
nat show resvmap <name> <interfacename>
```

19.14.2 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

19.14.3 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 global pool. To display global pool names, use the *nat list resvmaps* 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

19.14.4 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
```

19.14.5 See also

nat list resvmaps on page 297

security list interfaces on page 550

19.15 nat status

19.15.1 Syntax

```
nat status
```

19.15.2 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

19.15.3 Example

```
prompt> nat status

NAT enabled on:

  ID |    Name    | Interface  |  Type
-----------------------------------------
   1 | n2         | ip2        | internal
   2 | n1         | if1        | internal
-----------------------------------------
```

19.15.4 See also

nat enable on page 277

20.Port CLI commands

This chapter describes CLI support for ports.

20.1 Summary

20.1.1 Port CLI commands

The table below lists the port commands provided by the CLI:

Command	Reference
port ?	*port ?* on page 305
port set	*port set* on page 306
port show	*port show* on page 308

20.1.2 Port Console commands

The table below lists the *port* **console** commands and, if available, their equivalent CLI command:

Command	CLI Equivalent
bun list ports	Replaced by *port ?* on page 305
bun set port	Replaced by *port set* on page 306
bun show port	Replaced by *port show* on page 308

20.2 port ?

20.2.1 Syntax

```
port ?
```

20.2.2 Description

This command lists the ports that are configured in your GlobespanVirata ISOS system.

Ports are configured in the ISFS *initbun* configuration file for the product that you are using. For example, the initbun file for a PC-attached Gateway (USB) is located in:

atmos/products/usb-gateway/flashfs/initbun

20.2.3 Examples

The following examples lists the port names available on two different ISOS systems:

- BD6000 Series
- DM8010

BD6000 Series ports:

```
prompt> port ?
atm             Port name
ethernet        Port name
```

DM8010 ports:

```
prompt> port ?
hfa3841         Port name
ks8995e         Port name
realtek         Port name
```

20.2.4 See also

port set on page 306

port show on page 308

20.3 port set

20.3.1 Syntax

```
port <portname> set <attribute> <value>
```

20.3.2 Description

This command allows you to modify attributes on a 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 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 <portname> set*. For example:

```
prompt> port ethernet set ?
MaxMulticastListSize
MaxQueue
Reset
Disable
PromiscuousEnable
RxTestEnable
RxBroadcastEnable
RxMulticastEnable
RxMulticastAllEnable
RxAddressEnable
RxShortPktEnable
FastClockEnable
FullDuplexEnable
CrcEnable
PadShortDataEnable
Loopback
RxAnyEnable
HwFilterHigh
HwFilterLow
MAC
SnmpIfAdminStatus
SnmpIfLinkUpDownTrapEnable
SnmpIfPromiscuousMode
SnmpIfAlias
```

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

20.3.3 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
portname	The name of a port that is configured in your ISOS system. To display available ports, use the *port* command.	N/A
attribute	A single attribute of a specified 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

20.3.4 Example

```
prompt> port atm set DefaultPCR 3000
```

20.3.5 See also

port ? on page 305

port show on page 308

20.4 port show

20.4.1 Syntax

```
port <portname> show
```

20.4.2 Description

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

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

20.4.3 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
portname	The name of a port that is configured in your ISOS system. To display available ports, use the *port* command.	N/A

20.4.4 Example

```
prompt> port hfa3841 show
UpdateStats                    = 0
Version                        = 1.10
PortClassEthernet              = true
Reset                          = false
Disable                        = false
PromiscuousEnable              = false
RxBroadcastEnable              = true
RxMulticastEnable              = true
RxMulticastAllEnable           = true
RxAddressEnable                = true
FullDuplexEnable               = true
CrcEnable                      = true
MAC                            = 00:90:96:00:00:49
ApFirmwareVersion              =
PrimaryFirmwareVersion         =
ESSID                          = act1
```

```
DefaultChannel                  = 1
WepEncryption                   = disabled
DefaultTxKey                    = 0
Mode64Key0                      = 00-00-00-00-00
Mode64Key1                      = 00-00-00-00-00
Mode64Key2                      = 00-00-00-00-00
Mode64Key3                      = 00-00-00-00-00
Mode128Key0                     = 00-00-00-00-00-00-00-00-00-00-00-00-00
Mode128Key1                     = 00-00-00-00-00-00-00-00-00-00-00-00-00
Mode128Key2                     = 00-00-00-00-00-00-00-00-00-00-00-00-00
Mode128Key3                     = 00-00-00-00-00-00-00-00-00-00-00-00-00
RtsThreshold                    = 2432
FragmentationThreshold          = 2346
TxUnicastFrames                 = 0
TxMulticastFrames               = 0
TxFragments                     = 0
TxUnicastOctets                 = 0
TxMulticastOctets               = 0
TxDeferredTransmissions         = 0
TxSingleRetryFrames             = 0
TxMultipleRetryFrames           = 0
TxRetryLimitExceeded            = 0
TxDiscards                      = 0
RxUnicastFrames                 = 0
RxMulticastFrames               = 0
RxFragments                     = 0
RxUnicastOctets                 = 0
RxMulticastOctets               = 0
RxFCSErrors                     = 0
RxDiscardsNoBuffer              = 0
TxDiscardsWrongSA               = 0
RxDiscardsWEPUndecryptable      = 0
RxMessageInMsgFragments         = 0
RxMessageInBadMsgFragments      = 0
RxOk                            = 0
TxOk                            = 0
TxAborts                        = 0
RxCRCErrors                     = 0
TxCollisions                    = 0
RxOverlongPackets               = 0
TxExcessiveCollisions           = 0
```

```
RxShortPackets                  = 0
SnmpIfIndex                     = 2
SnmpIfDescr                     = BUN Ethernet device
SnmpIfType                      = 6
SnmpIfMtu                       = 1500
SnmpIfSpeed                     = 100000000
SnmpIfPhysAddress               = 00:90:96:00:00:49
SnmpIfOperStatus                = 1
SnmpIfAdminStatus               = 1
SnmpIfLastChange                = 218
SnmpIfInOctets                  = 0
SnmpIfInUcastPkts               = 0
SnmpIfInNUcastPkts              = 0
SnmpIfInDiscards                = 0
SnmpIfInErrors                  = 0
SnmpIfInUnknownProtos           = 0
SnmpIfOutOctets                 = 0
SnmpIfOutUcastPkts              = 0
SnmpIfOutNUcastPkts             = 0
SnmpIfOutDiscards               = 0
SnmpIfOutErrors                 = 0
SnmpIfOutQLen                   = 0
SnmpIfSpecific                  = (unset)
SnmpIfLinkUpDownTrapEnable      = 2
SnmpIfPromiscuousMode           = 2
SnmpIfAlias                     =
```

20.4.5 See also

port ? on page 305

port set on page 306

21.PPPoA CLI commands

This chapter describes the PPP over ATM CLI commands.

21.1 Summary

21.1.1 PPP CLI commands

The table below lists the PPP commands provided by the CLI:

Command	Description/Console command
pppoa add transport dialin pvc	*pppoa add transport dialin pvc* on page 317
pppoa add transport dialin svc	*pppoa add transport dialin svc* on page 319
pppoa add transport dialout pvc	*pppoa add transport dialout pvc* on page 320
pppoa add transport dialout svc	*pppoa add transport dialout svc* on page 322
pppoa clear transports	*pppoa clear transports* on page 323
pppoa delete transport	*pppoa delete transport* on page 324
pppoa list transports	*pppoa list transports* on page 325
pppoa set transport bt	*pppoa set transport bt* on page 327
pppoa set transport createroute	*pppoa set transport createroute* on page 328
pppoa set transport dialin pvc	*pppoa set transport dialin pvc* on page 329
pppoa set transport dialin svc	*pppoa set transport dialin svc* on page 331
pppoa set transport dialout pvc	*pppoa set transport dialout pvc* on page 332
pppoa set transport dialout svc	*pppoa set transport dialout svc* on page 334
pppoa set transport discoverdns primary	*pppoa set transport discoverdns primary* on page 335
pppoa set transport discoverdns secondary	*pppoa set transport discoverdns secondary* on page 337

Command	Description/Console command
pppoa set transport enabled|disabled	pppoa set transport enabled|disabled on page 339
pppoa set transport givedns client	pppoa set transport givedns client enabled|disabled on page 340
pppoa set transport givedns relay	pppoa set transport givedns relay enabled|disabled on page 342
pppoa set transport headers hdlc	pppoa set transport headers hdlc on page 344
pppoa set transport headers llc	pppoa set transport headers llc on page 346
pppoa set transport interface	pppoa set transport interface on page 349
pppoa set transport lcpechoevery	pppoa set transport lcpechoevery on page 350
pppoa set transport lcpmaxconf	pppoa set transport lcpmaxconf on page 351
pppoa set transport lcpmaxfail	pppoa set transport lcpmaxfail on page 352
pppoa set transport lcpmaxterm	pppoa set transport lcpmaxterm on page 353
pppoa set transport localip	pppoa set transport localip on page 354
pppoa set transport mbs	pppoa set transport mbs on page 356
pppoa set transport mcr	pppoa set transport mcr on page 357
pppoa set transport password	pppoa set transport password on page 358
pppoa set transport pcr	pppoa set transport pcr on page 359
pppoa set transport port	pppoa set transport port on page 360
pppoa set transport qosclass	pppoa set transport qosclass on page 361

Command	Description/Console command
pppoa set transport remotedns	*pppoa set transport remotedns* on page 363
pppoa set transport remoteip	*pppoa set transport remoteip* on page 365
pppoa set transport routemask	*pppoa set transport routemask* on page 366
pppoa set transport scr	*pppoa set transport scr* on page 367
pppoa set transport specificroute	*pppoa set transport specificroute* on page 368
pppoa set transport subnetmask	*pppoa set transport subnetmask* on page 370
pppoa set transport theylogin	*pppoa set transport theylogin* on page 371
pppoa set transport username	*pppoa set transport username* on page 373
pppoa set transport vci	*pppoa set transport vci* on page 374
pppoa set transport vpi	*pppoa set transport vpi* on page 375
pppoa set transport welogin	*pppoa set transport welogin* on page 376
pppoa show transport	*pppoa show transport* on page 378

21.1.2 PPP Console commands

The table below lists the *ppp* **console** commands and, if available, their equivalent CLI command:

Command	CLI Equivalent
ppp <channel> clear	Blacklisted command, see *<channel> clear* on page 840
ppp <channel> disable	Blacklisted command, see *<channel> disable* on page 841

Command	CLI Equivalent
ppp <channel> discard	Usable command, see *<channel> discard* on page 842
ppp <channel> echo	Usable command, see *<channel> echo* on page 843
ppp <channel> echo every	Usable command, see *<channel> echo every* on page 844
ppp <channel> enable	Blacklisted command, see *<channel> enable* on page 845
ppp <channel> event	Usable command, see *<channel> event* on page 846
ppp <channel> hdlc	Replaced by CLI command *pppoa set transport headers hdlc* on page 344, using the *hdlc* option.
ppp <channel> info	Replaced by CLI command *pppoa show transport* on page 378.
ppp <channel> interface	Replaced by CLI command *pppoa set transport interface* on page 349
ppp <channel> lcpmaxconfigure	Replaced by CLI command *pppoa set transport lcpmaxconf* on page 351
ppp <channel> lcpmaxfailure	Replaced by CLI command *pppoa set transport lcpmaxfail* on page 352
ppp <channel> lcpmaxterminate	Blacklisted command, see *<channel> lcpmaxterminate* on page 852
ppp <channel> llc	Replaced by CLI command *pppoa set transport headers llc* on page 346, using the *llc* option.
ppp <channel> pvc	Replaced by CLI command *pppoa set transport dialin pvc* on page 329 and *pppoa set transport dialout pvc* on page 332
ppp <channel> qos	Replaced by CLI command *pppoa set transport qosclass* on page 361, *pppoa set transport pcr* on page 359, *pppoa set transport bt* on page 327
ppp <channel> remoteip	Replaced by CLI command *pppoa set transport remoteip* on page 365.

Command	CLI Equivalent
ppp <channel> svc	Replaced by CLI command *pppoa set transport dialin svc* on page 331 and *pppoa set transport dialout svc* on page 334
ppp <channel> theylogin	Replaced by CLI command *pppoa set transport theylogin* on page 371
ppp <channel> tunnel	Blacklisted command, see *<channel> tunnel <n> <tunnel protocol> <dial direction>* on page 861
ppp <channel> welogin	Replaced by CLI command *pppoa set transport welogin* on page 376
ppp bcp	Replaced by CLI command *pppoa set transport headers hdlc* on page 344, using the *hdlc* option
ppp interface - localip	Replaced by CLI command *pppoa set transport localip* on page 354
ppp interface - stats	Replaced by CLI command *pppoa show transport* on page 378
ppp user	Blacklisted command, see *user* on page 866
ppp version	Replaced by CLI command *pppoa show transport* on page 378

21.2 pppoa add transport dialin pvc

21.2.1 Syntax

```
pppoa add transport <name> dialin pvc <interface> <port> <vpi> <vci>
```

21.2.2 Description

This command creates a PPPoA transport that accepts dialin connections over a PVC (Permanent Virtual Circuit). It allows you to specify the following information:

- the PPP interface to the channel that the PVC will use
- the ATM port that will transport data
- VPI (Virtual Path Identifier)
- VCI (Virtual Circuit Identifier)

The port/VPI/VCI combination must be unique for each transport.

21.2.3 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	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 PPPoA data. A single interface can be used by multiple channels. The interface value can be any positive integer.	N/A
port	The Virata system port that is used to transport ATM data.	N/A

vpi	A field in the ATM header. The VPI is used to identify the virtual path that a circuit belongs to. The VPI can be any value between 0 and 4095.	N/A
vci	Part of the ATM header. The VCI is a tag that identifies which channel a cell will travel over.The VCI can be any value between 1 and 65535.	N/A

21.2.4 Example

```
prompt> pppoa add transport pppoa1 dialin pvc 1 a1 0 800
```

21.2.5 See also

pppoa list transports on page 325

pppoa set transport vci on page 374

pppoa set transport vpi on page 375

To list ATM ports, see the usable console command BUN *list ports* on page 752.

21.3 pppoa add transport dialin svc

21.3.1 Syntax

```
pppoa add transport <name> dialin svc <interface>
```

21.3.2 Description

This command creates a PPPoA transport that accepts dialin connections over an SVC (Switched Virtual Circuit). It allows you to specify the PPP interface that SVC will use.

Note - You **must** have the *package_si* module included in your image build if you want to use SVCs.

21.3.3 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	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 PPPoA data. A single interface can be used by multiple channels. The interface value can be any positive integer.	N/A

21.3.4 Example

```
prompt> pppoa add transport pppoa2 dialin svc 2
```

21.3.5 See also

pppoa list transports on page 325

21.4 pppoa add transport dialout pvc

21.4.1 Syntax

```
pppoa add transport <name> dialout pvc <interface> <port> <vpi> <vci>
```

21.4.2 Description

This command creates a PPPoA transport that performs dialout over a PVC (Permanent Virtual Circuit). It allows you to specify the following PVC information:

- the PPP interface to the channel that the PVC will use
- the ATM port that will transport data
- VPI (Virtual Path Identifier)
- VCI (Virtual Circuit Identifier)

The port/VPI/VCI combination must be unique for each transport.

21.4.3 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	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 PPPoA data. A single interface can be used by multiple channels. The interface value can be any positive integer.	N/A
port	The Virata system port that is used to transport ATM data.	N/A

Option	Description	Default value
vpi	A field in the ATM header. The VPI is used to identify the virtual path that a circuit belongs to. The VPI can be any value between 0 and 4095.	N/A
vci	Part of the ATM header. The VCI is a tag that identifies which channel a cell will travel over.The VCI can be any value between 1 and 65535.	N/A

21.4.4 Example

```
prompt> pppoa add transport pppoa1 dialout pvc 1 a1 0 800
```

21.4.5 See also

pppoa list transports on page 325

To list ATM ports, see the usable console command BUN *list ports* on page 752.

21.5 pppoa add transport dialout svc

21.5.1 Syntax

```
pppoa add transport <name> dialout svc <interface> <atm address>
```

21.5.2 Description

This command creates a PPPoA transport that performs dialout over an SVC (Switched Virtual Circuit). It allows you to specify the PPP interface that SVC will use and the ATM address.

Note - You **must** have the *package_si* module included in your image build if you want to use SVCs.

21.5.3 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	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 PPPoA data. A single interface can be used by multiple channels. The interface value can be any positive integer.	N/A
ATM address	The ATM address of the remote server that you want to use.	N/A

21.5.4 Example

```
prompt> pppoa add transport pppoa3 dialout svc 3
47.00.83.10.a2.b1.00.00.00.00.00.00.00.20.2b.01.00.07.00
```

21.5.5 See also

pppoa list transports on page 325

21.6 pppoa clear transports

21.6.1 Syntax

```
pppoa clear transports
```

21.6.2 Description

This command deletes all PPPoA transports that were created using the *pppoa add transport* commands.

21.6.3 Example

```
prompt> pppoa clear transports
```

21.6.4 See also

pppoa delete transport on page 324

21.7 pppoa delete transport

21.7.1 Syntax

```
pppoa delete transport {<name>|<number>}
```

21.7.2 Description

This command deletes a single PPPoA transport. The PVC or SVC attached to the transport is also deleted.

21.7.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A

21.7.4 Example

```
prompt> pppoa delete transport pppoa1
```

21.7.5 See also

pppoa list transports on page 325

21.8 pppoa list transports

21.8.1 Syntax

```
pppoa list transports
```

21.8.2 Description

This command lists PPPoA transports that have been created using the *pppoa add transport* commands. It displays the following information about the transports:

- transport identification number
- transport name
- ATM port used (if applicable)
- Virtual Circuit Identifier (VCI) used (if applicable)
- Virtual Path Identifier (VPI) used (if applicable)

21.8.3 Example

```
prompt> pppoa list transports
PPPOA transports:

 ID  |    Name    | Port      | Vci       | Vpi
-----|------------|-----------|-----------|-----------
   1 | p2         | N/A       | N/A       | N/A
   2 | p1         | a1        | 800       | 0
-------------------------------------------------------
```

21.8.4 See also

pppoa show transport on page 378

21.9 pppoa set transport autoconnect

21.9.1 Syntax

```
pppoa set transport {<name>|<number>} autoconnect {enabled|disabled
```

21.9.2 Description

This command allows you to enable/disable the PPPoA autoconnect function. If enabled, PPPoA automatically connects to TCP/IP whenever a user requests TCP/IP packets from a public destination.

21.9.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
enabled	Enables PPPoA autoconnect.	disabled
disabled	Disables PPPoA autoconnect.	

21.9.4 Example

```
prompt> pppoa set transport pppoa1 autoconnect enabled
```

21.9.5 See also

pppoa list transports on page 325

21.10 pppoa set transport bt

21.10.1 Syntax

```
pppoa set transport {<name>|<number>} bt <burst tolerance>
```

21.10.2 Description

This command applies to existing PVC transports - it does not apply to SVC transports. This command sets the burst tolerance (bt) for an existing PPPoA transport. This command is only valid if you set VBR or VBR RT as the QoS Class using the *pppoa set transport qosclass* command.

21.10.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
burst tolerance	Controls the duration of traffic bursts on VBR (Variable Bit Rate) and VBR RT (VBR Real Time) channels. This value overrides an existing MBS value (if set). The BT can be any value between 0 and 100.	0

21.10.4 Example

```
prompt> pppoa set transport pppoa1 bt 5
```

21.10.5 See also

pppoa set transport mbs on page 356

21.11 pppoa set transport createroute

21.11.1 Syntax

```
pppoa set transport {<name>|<number>} createroute {enabled|disabled}
```

21.11.2 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 route, use the *ip list routes* command. The route is removed when the PPP link is disconnected.

21.11.3 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 PPPoA transport. To display transport names, use the *pppoa 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.	

21.11.4 Example

```
prompt> pppoa set transport pppoa1 createroute disabled
```

21.11.5 See also

pppoa show transport on page 378

pppoa set transport specificroute on page 368

ip list routes on page 641

21.12 pppoa set transport dialin pvc

21.12.1 Syntax

```
pppoa set transport {<name>|<number>} dialin pvc <port> <vpi> <vci>
```

21.12.2 Description

This command sets an existing PPPoA transport to accept dialin connections over a PVC (Permanent Virtual Circuit). This replaces the transports existing dialin/dialout setting over PVC/SVC. The PVC uses the interface that was specified when the transport was created.

The command allows you to specify the following PVC information:

- the ATM port that will transport data
- VPI (Virtual Path Identifier)
- VCI (Virtual Circuit Identifier)

The port/VPI/VCI combination must be unique for each transport.

21.12.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
port	The Virata system port that is used to transport ATM data.	N/A

vpi	A field in the ATM header. The VPI is used to identify the virtual path that a circuit belongs to. The VPI can be any value between 0 and 4095.	N/A
vci	Part of the ATM header. The VCI is a tag that identifies which channel a cell will travel over.The VCI can be any value between 1 and 65535.	N/A

21.12.4 Example

```
prompt> pppoa set transport pppoa2 dialin pvc a1 0 800
```

21.12.5 See also

pppoa list transports on page 325

pppoa set transport dialout pvc on page 332

To list ATM ports, see the usable console command BUN *list ports* on page 752.

21.13 pppoa set transport dialin svc

21.13.1 Syntax

```
pppoa set transport {<name>|<number>} dialin svc
```

21.13.2 Description

This command sets an existing PPPoA transport to accept dialin connections over an SVC (Switched Virtual Circuit). This replaces the transports existing dialin/dialout setting over PVC/SVC. The SVC uses the interface that was specified when the transport was created.

Note - You **must** have the *package_si* module included in your image build if you want to use SVCs.

21.13.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A

21.13.4 Example

```
prompt> pppoa set transport pppoa1 dialin svc
```

21.13.5 See also

pppoa list transports on page 325

pppoa set transport dialout svc on page 334

21.14 pppoa set transport dialout pvc

21.14.1 Syntax

```
pppoa set transport {<name>|<number>} dialout pvc <port> <vpi> <vci>
```

21.14.2 Description

This command sets a PPPoA transport to perform dialout over a PVC (Permanent Virtual Circuit). This replaces the transports existing dialin/dialout setting over PVC/SVC. The PVC uses the interface that was specified when the transport was created.

The command allows you to specify the following PVC information:

- the ATM port that will transport data
- VPI (Virtual Path Identifier)
- VCI (Virtual Circuit Identifier)

The port/VPI/VCI combination must be unique for each transport.

21.14.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
port	The Virata system port that is used to transport ATM data.	N/A

vpi	A field in the ATM header. The VPI is used to identify the virtual path that a circuit belongs to. The VPI can be any value between 0 and 4095.	N/A
vci	Part of the ATM header. The VCI is a tag that identifies which channel a cell will travel over.The VCI can be any value between 1 and 65535.	N/A

21.14.4 Example

```
prompt> pppoa set transport pppoa2 dialout pvc a1 0 800
```

21.14.5 See also

pppoa list transports on page 325

pppoa set transport dialin pvc on page 329

To list ATM ports, see the usable console command BUN *list ports* on page 752.

21.15 pppoa set transport dialout svc

21.15.1 Syntax

```
pppoa set transport {<name>|<number>} dialout svc
```

21.15.2 Description

This command sets an existing PPPoA transport to perform dialout over an SVC (Switched Virtual Circuit). This replaces the transport's existing dialin/dialout setting over PVC/SVC. The SVC uses the interface that was specified when the transport was created.

Note - You **must** have the *package_si* module included in your image build if you want to use SVCs.

21.15.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A

21.15.4 Example

```
prompt> pppoa set transport pppoa3 dialout svc
```

21.15.5 See also

pppoa list transports on page 325

pppoa set transport dialin svc on page 331

21.16 pppoa set transport discoverdns primary

21.16.1 Syntax

```
pppoa set transport {<name>|<number>} discoverdns primary
{enabled|disabled}
```

21.16.2 Description

Note - You must enable one of the *pppoa set transport givedns* commands in order for this command setting to work. See *pppoa set transport givedns client enabled|disabled* on page 340, or *pppoa set transport givedns relay enabled|disabled* on page 342

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 *pppoa set transport givedns* commands is also enabled.

21.16.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa 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.	

21.16.4 Example

```
prompt> pppoa set transport pppoa3 discoverdns primary
enabled
```

21.16.5 See also

pppoa set transport discoverdns secondary on page 337

pppoa set transport givedns client enabled|disabled on page 340

pppoa set transport givedns relay enabled|disabled on page 342

pppoa set transport remotedns on page 363

21.17 pppoa set transport discoverdns secondary

21.17.1 Syntax

```
pppoa set transport {<name>|<number>} discoverdns
secondary {enabled|disabled}
```

21.17.2 Description

Note - You must enable one of the *pppoa set transport givedns* commands in order for this command setting to work. See *pppoa set transport givedns client enabled|disabled* on page 340, or *pppoa set transport givedns relay enabled|disabled* on page 342

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 *pppoa set transport givedns* commands is also enabled.

21.17.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
enabled	A secondary DNS server IP address is requested.	enabled
disabled	A secondary DNS server IP address is not requested.	

21.17.4 Example

```
prompt> pppoa set transport pppoa3 discoverdns secondary
enabled
```

21.17.5 See also

pppoa set transport discoverdns primary on page 335

pppoa set transport givedns client enabled|disabled on page 340

pppoa set transport givedns relay enabled|disabled on page 342

pppoa set transport remotedns on page 363

21.18 pppoa set transport enabled|disabled

21.18.1 Syntax

```
pppoa set transport {<name>|<number>} {enabled|disabled}
```

21.18.2 Description

This command explicitly enables/disables a PPPoA 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.

21.18.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
enabled	Enables a PPPoA transport.	disabled
disabled	Disables a PPPoA transport.	

21.18.4 Example

```
prompt> pppoa set transport pppoa1 enabled
```

21.18.5 See also

pppoa list transports on page 325

21.19 pppoa set transport givedns client enabled|disabled

21.19.1 Syntax

```
pppoa set transport {<name>|<number>} givedns client {enabled | disabled}
```

21.19.2 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.

21.19.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
enabled	IPCP can request a DNS server IP address and then give the address to DNS client.	disabled
disabled	IPCP cannot request a DNS server IP address and then give the address to DNS client.	

21.19.4 Example

```
prompt> pppoa set transport pppoa1 givedns client enabled
```

21.19.5 See also

pppoa set transport givedns relay enabled|disabled on page 342

pppoa set transport remotedns on page 363

pppoa set transport discoverdns primary on page 335

pppoa set transport discoverdns secondary on page 337

DNS Client CLI commands on page 145

For more information on DNS client, see *ATMOS DNS Client Functional Specification: DO-008322-PS*.

For information on including processes in your image build, see the Software User's Guide for the Virata system that you are using.

For information on DNS implementation and specification, see *http://www.ietf.org/rfc/rfc1035.txt*.

21.20 pppoa set transport givedns relay enabled|disabled

21.20.1 Syntax

```
pppoa set transport {<name>|<number>} givedns relay {enabled | disabled}
```

21.20.2 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.

21.20.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing pppoa transport. To display transport numbers, use the *pppoa list transports* command.	N/A
enabled	IPCP can request a DNS server IP address and then give the address to DNS relay.	disabled
disabled	IPCP cannot request a DNS server IP address and then give the address to DNS relay.	

21.20.4 Example

```
prompt> pppoa set transport pppoa1 givedns relay enabled
```

21.20.5 See also

pppoa set transport givedns client enabled|disabled on page 340

pppoa set transport remotedns on page 363

pppoa set transport discoverdns primary on page 335

pppoa set transport discoverdns secondary on page 337

DNS Relay CLI commands on page 155

For more information on DNS relay, see *Virata DNS Relay Functional Specification: DO-007692-PS*.

For information on including processes in your image build, see the Software User's Guide for the Virata system that you are using.

For information on DNS implementation and specification, see *http://www.ietf.org/rfc/rfc1035.txt*.

21.21 pppoa set transport headers hdlc

21.21.1 Syntax

```
pppoa set transport {<name>|<number>} headers hdlc {enabled|disabled}
```

21.21.2 Description

This command allows you to enable/disable whether your system can transmit and receive packets containing HDLC headers. If you want LLC packets to be transmitted and received instead of/as well as HDLC packets, use the *pppoa set transport headers llc enabled* command.

When both HDLC and LLC headers are disabled, the default encapsulation method is VC multiplexed (VC Mux). PPP determines which format to use to transmit/receive packets by 'learning' the format information from incoming packet headers.

21.21.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
enabled	Packets that have HDLC headers can be transmitted/received.	disabled (if LLC headers are disabled too, the default value is VC Mux)
disabled	Packets that have HDLC headers can not be transmitted/received.	

21.21.4 Example

```
prompt> pppoa set transport pppoa1 headers hdlc enabled
```

21.21.5 See also

pppoa list transports on page 325

pppoa show transport on page 378

pppoa set transport headers llc on page 346

21.22 pppoa set transport headers llc

21.22.1 Syntax

```
pppoa set transport {<name>|<number>} headers llc {enabled|disabled}
```

21.22.2 Description

This command allows you to enable/disable whether your system can transmit and receive packets containing LLC headers. If you want HDLC packets to be transmitted and received instead of/as well as LLC packets, use the *pppoa set transport headers hdlc enabled* command.

When both LLC and HDLC headers are disabled, the default encapsulation method is VC multiplexed (VC Mux). PPP determines which format to use to transmit/receive packets by 'learning' the format information from incoming packet headers.

21.22.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoe list transports* command.	N/A
enabled	Packets that have LLC headers can be transmitted/received.	disabled (if HDLC headers are disabled too, the default value is VC Mux)
disabled	Packets that have LLC headers can not be transmitted/received.	

21.22.4 Example

```
prompt> pppoa set transport pppoa1 headers llc enabled
```

21.22.5 See also

pppoa list transports on page 325

pppoa show transport on page 378

pppoa set transport headers hdlc on page 344

21.23 pppoa set transport idletimeout

21.23.1 Syntax

```
pppoa set transport {<name>|<number>} idletimeout <idletimeout>
```

21.23.2 Description

This command allows you to set an ‘idle’ time out for your LAN connection. If you are connected to an ISP via PPPoA but fail to send a request for data within a specified time limit, the PPPoA session is disabled.

21.23.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
idletimeout	The length of time (in minutes) that a PPPoA session connected to an ISP can remain idle before the session is disabled. The time can be any value between 0 and 60. A value of 0 means that no idletimeout is set.	0

21.23.4 Example

```
prompt> pppoa set transport pppoa1 idletimeout 20
```

21.23.5 See also

pppoa list transports on page 325

pppoa set transport lcpechoevery on page 350

21.24 pppoa set transport interface

21.24.1 Syntax

```
pppoa set transport {<name>|<number>} interface <interface>
```

21.24.2 Description

This command sets the PPP interface for an existing PPPoA transport.

21.24.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
interface	The PPP interface to a channel that transports PPPoA data. A single interface can be used by multiple channels. The interface value can be any positive integer.	N/A

21.24.4 Example

```
prompt> pppoa set transport pppoa2 interface 4
```

21.24.5 See also

pppoa show transport on page 378

pppoa list transports on page 325

21.25 pppoa set transport lcpechoevery

21.25.1 Syntax

```
pppoa set transport {<name>|<number>} lcpechoevery <interval>
```

21.25.2 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.

21.25.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa 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

21.25.4 Example

```
prompt> pppoa set transport pppoa2 lcpechoevery 0
```

21.25.5 See also

pppoa show transport on page 378

pppoa list transports on page 325

21.26 pppoa set transport lcpmaxconf

21.26.1 Syntax

```
pppoa set transport {<name>|<number>} lcpmaxconf <lcp max configure>
```

21.26.2 Description

This command sets the Link Control Protocol (LCP) maximum configure number for an existing PPPoA transport.

21.26.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa 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

21.26.4 Example

```
prompt> pppoa set transport pppoa1 lcpmaxconf 20
```

21.26.5 See also

pppoa show transport on page 378

pppoa list transports on page 325

21.27 pppoa set transport lcpmaxfail

21.27.1 Syntax

```
pppoa set transport {<name>|<number>} lcpmaxfail <lcp max fail>
```

21.27.2 Description

This command sets the Link Control Protocol (LCP) maximum fail number for an existing PPPoA transport.

21.27.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa 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

21.27.4 Example

```
prompt> pppoa set transport pppoa1 lcpmaxfail 20
```

21.27.5 See also

pppoa show transport on page 378

pppoa list transports on page 325

21.28 pppoa set transport lcpmaxterm

21.28.1 Syntax

```
pppoa set transport {<name>|<number>} lcpmaxterm <lcp max terminate>
```

21.28.2 Description

This command sets the Link Control Protocol (LCP) maximum terminate number for an existing PPPoA transport.

21.28.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
lcp 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

21.28.4 Example

```
prompt> pppoa set transport pppoa1 lcpmaxterm 20
```

21.28.5 See also

pppoa show transport on page 378

pppoa list transports on page 325

21.29 pppoa set transport localip

21.29.1 Syntax

```
pppoa set transport {<name>|<number>} localip <ip-address>
```

21.29.2 Description

This command is only applicable to dialin SVC or PVC 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.

Note - You **must** have the *package_si* module included in your image build if you want to use SVCs.

21.29.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa 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

21.29.4 Example

```
prompt> pppoa set transport pppoa1 localip 192.168.103.2
```

21.29.5 See also

pppoa show transport on page 378

pppoa list transports on page 325

pppoa set transport remoteip on page 365

21.30 pppoa set transport mbs

21.30.1 Syntax

```
pppoa set transport {<name>|<number>} mbs <max burst size>
```

21.30.2 Description

This command applies to existing PVC transports - it does not apply to SVC transports. It sets the maximum burst size (mbs) for the PPPoA transport.

This command is only valid if you set VBR or VBR RT as the QoS Class using the *pppoa set transport quosclass* command.

21.30.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
maximum burst size	Controls the maximum burst size for VBR (Variable Bit Rate) and VBR RT (VBR Real Time) channels. This value overrides an existing BT value (if set). The MBS can be any value between 0 and 100.	0

21.30.4 Example

```
prompt> pppoa set transport pppoa3 mbs 10
```

21.30.5 See also

pppoa set transport bt on page 327

pppoa set transport qosclass on page 361

21.31 pppoa set transport mcr

21.31.1 Syntax

```
pppoa set transport {<name>|<number>} mcr <min cell rate>
```

21.31.2 Description

This command applies to existing PVC transports - it does not apply to SVC transports. This command sets the minimum cell rate for an existing PPPoA transport.

21.31.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
minimum cell rate	Determines the minimum rate at which ATM cells are allowed to be sent along the PPPoA transport.	0

21.31.4 Example

```
prompt> pppoa set transport pppoa2 mcr 0
```

21.31.5 See also

pppoa set transport pcr on page 359

21.32 pppoa set transport password

21.32.1 Syntax

```
pppoa set transport {<name>|<number>} password <password>
```

21.32.2 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.

21.32.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa 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 *pppoa show transport* command.	N/A

21.32.4 Example

```
prompt> pppoa set transport pppoa2 password mercury
```

21.32.5 See also

pppoa list transports on page 325

pppoa show transport on page 378

pppoa set transport username on page 373

21.33 pppoa set transport pcr

21.33.1 Syntax

```
pppoa set transport {<name>|<number>} pcr <peak cell rate>
```

21.33.2 Description

This command applies to existing PVC transports - it does not apply to SVC transports. This command sets the peak cell rate (pcr) for an existing PPPoA transport.

21.33.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
peak cell rate	Determines the maximum rate at which ATM cells are allowed to be sent along the PPPoA transport. The PCR can be any value from 0 up to the maximum PortSpeed parameter set when the port was created (using the initbun file in FlashFS or the console command BUN *set port* on page 754).	0

21.33.4 Example

```
prompt> pppoa set transport pppoa2 pcr 50000
```

21.33.5 See also

pppoa set transport mcr on page 357

21.34 pppoa set transport port

21.34.1 Syntax

```
pppoa set transport {<name>|<number>} port <port>
```

21.34.2 Description

This command applies to existing PVC transports - it does not apply to SVC transports. This command sets the port that an existing transport uses to transport PPPoA data.

The port/VPI/VCI combination must be unique for each transport.

21.34.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
port	The Virata system port that is used to transport ATM data.	N/A

21.34.4 Example

```
prompt> pppoa set transport pppoa4 port a1
```

21.34.5 See also

pppoa list transports on page 325

To list ATM ports, see the usable console command BUN *list ports* on page 752.

21.35 pppoa set transport qosclass

21.35.1 Syntax

```
pppoa set transport {<name>|<number>} qosclass {ubr|cbr|vbr|vbrrt|abr|qfc}
```

21.35.2 Description

This command applies to existing PVC transports - it does not apply to SVC transports. It sets the quality of service class for the transport.

21.35.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A

Option	Description	Default value
ubr	Unspecified Bit Rate; non-constant and unpredictable data transport rate. PCR (Peak Cell Rate) is the average and maximum speed of transmission.	UBR
cbr	Constant Bit Rate; constant demand and predictable data transport rate. PCR is the average and maximum speed of transmission.	
vbr	Variable Bit Rate; non-constant but predictable data transport rate that uses Non-Real-Time (NRT). You can specify the PCR, SCR, BT and MBS for VBR traffic.	
vbrrt	Variable Bit Rate Real-Time; non-constant but predictable data transport rate that uses Real-Time (RT). You can specify the PCR, SCR, BT and MBS for VBRRT traffic.	
abr	Available Bit Rate; non-constant and unpredictable data transport rate that provides ATM-layer feedback and flow control.	
qfc	Quantum Flow Control; ATM flow control protocol that supports ABR.	

21.35.4 Example

```
prompt> pppoa set transport pppoa3 qosclass abr
```

21.35.5 See also

pppoa show transport on page 378

pppoa set transport qosclass on page 361

21.36 pppoa set transport remotedns

21.36.1 Syntax

```
pppoa set transport {<name>|<number>} remotedns
<ipaddress> [<ipaddress2>]
```

21.36.2 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*.

21.36.3 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 pppoa transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing pppoa transport. To display transport numbers, use the *pppoa 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)

21.36.4 Examples

Example One - setting a primary address

```
prompt> pppoa set transport pppoa1 remotedns 192.168.102.3
```

Example Two - setting primary and secondary addresses

To set primary and secondary addresses, use this command syntax:

```
prompt> pppoa set transport pppoa1 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> pppoa set transport pppoa1 remotedns 192.168.102.3
0.0.0.0
```

21.36.5 See also

pppoa set transport givedns client enabled|disabled on page 340

pppoa set transport givedns relay enabled|disabled on page 342

pppoa set transport discoverdns primary on page 335

pppoa set transport discoverdns secondary on page 337

21.37 pppoa set transport remoteip

21.37.1 Syntax

```
pppoa set transport {<name>|<number>} remoteip <ip-address>
```

21.37.2 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.

21.37.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
ip-address	The ip address of the local 'server-end' of an interface displayed in the following format: 111.222.254.4	0.0.0.0

21.37.4 Example

```
prompt> pppoa set transport pppoa1 remoteip 192.168.103.2
```

21.37.5 See also

pppoa show transport on page 378

pppoa list transports on page 325

pppoa set transport localip on page 354

21.38 pppoa set transport routemask

21.38.1 Syntax

```
pppoa set transport {<name>|<number>} routemask <mask>
```

21.38.2 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.

21.38.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa 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

21.38.4 Example

```
prompt> pppoa set transport pppoa1 routemask 0.0.0.0
```

21.38.5 See also

pppoa show transport on page 378

pppoa list transports on page 325

21.39 pppoa set transport scr

21.39.1 Syntax

```
pppoa set transport {<name>|<number>} scr <sustainable cell rate>
```

21.39.2 Description

This command applies to existing PVC transports - it does not apply to SVC transports. This command is only valid if you set VBR or VBR RT as the QoS Class using the *pppoa set transport qosclass* command.

21.39.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
sustainable cell rate	Sustainable Cell Rate; the average cell rate for a VBR or VBR RT connection. The SCR can be any positive value that is less than the PortSpeed and the PCR for the channel.	0

21.39.4 Example

```
prompt> pppoa set transport pppoa2 scr 25000
```

21.39.5 See also

pppoa set transport qosclass on page 361

21.40 pppoa set transport specificroute

21.40.1 Syntax

```
pppoa set transport {<name>|<number>} specificroute {enabled | disabled}
```

21.40.2 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 *pppoa 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 *pppoa set transport createroute* command is set to *disabled*.

21.40.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa 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.	

21.40.4 Example

```
prompt> pppoa set transport pppoa1 specificroute disabled
```

21.40.5 See also

pppoa set transport routemask on page 366

pppoa set transport createroute on page 328

pppoa list transports on page 325

21.41 pppoa set transport subnetmask

21.41.1 Syntax

```
pppoa set transport {<name>|<number>} subnetmask <mask>
```

21.41.2 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.

21.41.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
mask	The subnet mask used by the local IP interface connected to the PPP transport.	0.0.0.0

21.41.4 Example

```
prompt> pppoa set transport pppoa1 subnetmask 255.255.255.0
```

21.41.5 See also

pppoa list transports on page 325

21.42 pppoa set transport theylogin

21.42.1 Syntax

```
pppoa set transport {<name>|<number>} theylogin {none|pap|chap}
```

21.42.2 Description

This command sets the authentication method that remote PPP clients must use to dialin to the server. 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.

21.42.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa 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.	

21.42.4 Example

```
prompt> pppoa set transport pppoa2 theylogin pap
```

21.42.5 See also

pppoa list transports on page 325

pppoa show transport on page 378

system add user on page 585

system set user maydialin on page 611

21.43 pppoa set transport username

21.43.1 Syntax

```
pppoa set transport {<name>|<number>} username <username>
```

21.43.2 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.

21.43.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa 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 digits. To display the username, use the *pppoa show transport* command.	N/A

21.43.4 Example

```
prompt> pppoa set transport pppoa2 username jsmith
```

21.43.5 See also

pppoa set transport password on page 358

21.44 pppoa set transport vci

21.44.1 Syntax

```
pppoa set transport {<name>|<number>} vci <vci>
```

21.44.2 Description

This command applies to existing PVC transports - it does not apply to SVC transports.This command sets the Virtual Circuit Identifier.

21.44.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
vci	Part of the ATM header. The VCI is a tag that identifies which channel a cell will travel over. The VCI can be any value between 1 and 65535.	N/A

21.44.4 Example

```
prompt> pppoa set transport pppoa4 vci 800
```

21.44.5 See also

pppoa list transports on page 325

pppoa show transport on page 378

pppoa set transport vpi on page 375

21.45 pppoa set transport vpi

21.45.1 Syntax

```
pppoa set transport {<name>|<number>} vpi <vpi>
```

21.45.2 Description

This command applies to existing PVC transports - it does not apply to SVC transports.This command sets the Virtual Path Identifier.

21.45.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
vpi	A field in the ATM header. The VPI is used to identify the virtual path that a circuit belongs to. The VPI can be any value between 0 and 4095.	N/A

21.45.4 Example

```
prompt> pppoa set transport pppoa3 vpi 0
```

21.45.5 See also

pppoa list transports on page 325

pppoa show transport on page 378

pppoa set transport vci on page 374

21.46 pppoa set transport welogin

21.46.1 Syntax

```
pppoa set transport {<name>|<number>} welogin {none|auto|pap|chap}
```

21.46.2 Description

This command sets the authentication protocol used to connect to external PPP servers (dial-out).

21.46.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A
none	No authentication method is used.	None
auto	The authentication protocol used by the remote PPP server is discovered and used.	
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.	

21.46.4 Example

```
prompt> pppoa set transport pppoa2 theylogin pap
```

21.46.5 See also

pppoa set transport theylogin on page 371

pppoa show transport on page 378

pppoa list transports on page 325

21.47 pppoa show transport

21.47.1 Syntax

```
pppoa show transport {<name>|<number>}
```

21.47.2 Description

This command displays the following information about an existing PPPoA transport:

- Description
- Summary - the connection state
- Server - dialin status
- Headers - the data format that the transport can accept or receive
- SVC status (true or false)

- Local IP address
- Subnet mask
- Remote IP address
- Remote DNS
- 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 Echo Every

- ATM address (for SVC transports only)
- Auto-connect status
- Idletime status

- ATM port (for PVC transports only)
- Rx VPI (for PVC transports only)
- Rx VCI (for PVC transports only)

- Quality of Service (QoS) class (for PVC transports only)
- Burst tolerance (for PVC transports only)
- Sustainable Cell Rate (SCR) (for PVC transports only)
- Maximum burst size (MBS) (for PVC transports only)
- Maximum Cell Rate (MCR) (for PVC transports only)

21.47.3 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 PPPoA transport. To display transport names, use the *pppoa list transports* command.	N/A
number	A number that identifies an existing PPPoA transport. To display transport numbers, use the *pppoa list transports* command.	N/A

21.47.4 Example

There are two examples given below. Example one is of an SVC transport. Example two is of a PVC transport.

Example 1 - SVC

```
prompt> pppoa show transport pppoa1
PPP Transport: pppoa1
Description : pppoa1
                Summary : disabled
```

```
                Server : true
          Headers: learn             SVC: true

             Local Ip : 192.168.100.1
           Subnet Mask : 255.255.255.0
            Remote Ip : 192.168.100.2
           Remote DNS : N/A
      Give DNSto Client : true
       Give DNSto Relay : true

          Create Route : true
         Specific Route : false
             Route Mask : 255.0.0.0

       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
          Auto Connect : false
          Idle Timeout : 0
ATM address: 47.00.83.10.a2.b1.00.00.00.00.00.00.00.00.20.2b.01.00.07.00
```

Example 2 - PVC

```
prompt> pppoa show transport pppoa2
PPP Transport: pppoa2
Summary : enabled, down
                Server : true
          Headers: learn             SVC: false

             Local Ip : 192.168.100.1
           Subnet Mask : 255.255.255.0
            Remote Ip : 192.168.100.2
           Remote DNS : N/A
      Give DNSto Client : true
       Give DNSto Relay : true
```

```
                Create Route : true
              Specific Route : false
                  Route Mask : 255.0.0.0

            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
                Auto Connect : false
                Idle Timeout : 0

                        Port : a1
                     Rx Vpi : N/A
                     Rx Vci : 100

                       Class : UBR
             Burst Tolerance : N/A
       Sustainable Cell Rate : N/A
                         MBS : N/A
                         MCR : N/A
```

21.47.5 See also

pppoa list transports on page 325

22.PPPoE CLI commands

This chapter describes the PPP over Ethernet CLI commands.

22.1 Summary

22.1.1 PPPoE CLI commands

The table below lists the PPPoE commands provided by the CLI:

Command	Description/Console command
pppoe add transport dialout pvc	*pppoe add transport dialout pvc* on page 389
pppoe add transport dialout eth	*pppoe add transport dialout eth* on page 391
pppoe clear transports	*pppoe clear transports* on page 393
pppoe delete transport	*pppoe delete transport* on page 394
pppoe list transports	*pppoe list transports* on page 395
pppoe set transport enabled|disabled	*pppoe set transport enabled|disabled* on page 407
pppoe set transport autoconnect enabled|disabled	*pppoe set transport autoconnect* on page 398
pppoe set transport bt	*pppoe set transport bt* on page 399
pppoe set transport createroute	*pppoe set transport createroute* on page 400
pppoe set transport discoverdns primary	*pppoe set transport discoverdns primary* on page 402
pppoe set transport discoverdns secondary	*pppoe set transport discoverdns secondary* on page 404
pppoe set transport eth	*pppoe set transport eth* on page 406
pppoe set transport enabled|disabled	*pppoe set transport enabled|disabled* on page 407
pppoe set transport givedns client	*pppoe set transport givedns client enabled|disabled* on page 408
pppoe set transport givedns relay	*pppoe set transport givedns relay enabled|disabled* on page 410

Command	Description/Console command
pppoe set transport headers hdlc	*pppoe set transport headers hdlc* on page 412
pppoe set transport headers llc	*pppoe set transport headers llc* on page 414
pppoe set transport interface	*pppoe set transport interface* on page 417
pppoe set transport idletimeout	*pppoe set transport idletimeout* on page 416
pppoe set transport lcpechoevery	*pppoe set transport lcpechoevery* on page 418
pppoe set transport lcpmaxconf	*pppoe set transport lcpmaxconf* on page 419
pppoe set transport lcpmaxfail	*pppoe set transport lcpmaxfail* on page 420
pppoe set transport lcpmaxterm	*pppoe set transport lcpmaxterm* on page 421
pppoe set transport localip	*pppoe set transport localip* on page 422
pppoe set transport mbs	*pppoe set transport mbs* on page 423
pppoe set transport mcr	*pppoe set transport mcr* on page 424
pppoe set transport password	*pppoe set transport password* on page 425
pppoe set transport pcr	*pppoe set transport pcr* on page 426
pppoe set transport port	*pppoe set transport port* on page 427
pppoe set transport qosclass	*pppoe set transport qosclass* on page 428
pppoe set transport remotedns	*pppoe set transport remotedns* on page 430
pppoe set transport remoteip	*pppoe set transport remoteip* on page 432

Command	Description/Console command
pppoe set transport routemask	*pppoe set transport routemask* on page 433
pppoe set transport scr	*pppoe set transport scr* on page 434
pppoe set transport specificroute	*pppoe set transport specificroute* on page 437
pppoe set transport subnetmask	*pppoe set transport subnetmask* on page 439
pppoe set transport servicename	*pppoe set transport servicename* on page 435
pppoe set transport theylogin	*pppoe set transport theylogin* on page 440
pppoe set transport username	*pppoe set transport username* on page 442
pppoe set transport vci	*pppoe set transport vci* on page 443
pppoe set transport vpi	*pppoe set transport vpi* on page 444
pppoe set transport welogin	*pppoe set transport welogin* on page 445
pppoe show transport	*pppoe show transport* on page 447

22.1.2 PPP Console commands

The table below lists the *ppp* **console** commands and, if available, their equivalent CLI command:

Command	CLI Equivalent
ppp <channel> clear	Blacklisted command, see *<channel> clear* on page 850
ppp <channel> disable	Blacklisted command, see *<channel> disable* on page 851
ppp <channel> discard	Usable command, see *<channel> discard* on page 852

Command	CLI Equivalent
ppp <channel> echo	Usable command, see *<channel> echo* on page 853
ppp <channel> echo every	Usable command, see *<channel> echo every* on page 854
ppp <channel> enable	Blacklisted command, see *<channel> enable* on page 855
ppp <channel> event	Usable command, see *<channel> event* on page 856
ppp <channel> hdlc	Replaced by CLI command *pppoe set transport headers hdlc* on page 412, using the *hdlc* option.
ppp <channel> info	Replaced by CLI command *pppoe show transport* on page 447.
ppp <channel> interface	Replaced by CLI command *pppoe set transport interface* on page 417
ppp <channel> lcpmaxconfigure	Replaced by CLI command *pppoe set transport lcpmaxconf* on page 419
ppp <channel> lcpmaxfailure	Replaced by CLI command *pppoe set transport lcpmaxfail* on page 420
ppp <channel> lcpmaxterminate	Blacklisted command, see *<channel> lcpmaxterminate* on page 862
ppp <channel> llc	Replaced by CLI command *pppoe set transport headers llc* on page 414, using the *llc* option.
ppp <channel> qos	Replaced by CLI command *pppoe set transport qosclass* on page 428, *pppoe set transport pcr* on page 426, *pppoe set transport bt* on page 399
ppp <channel> remoteip	Replaced by CLI command *pppoe set transport remoteip* on page 432.
ppp <channel> theylogin	Replaced by CLI command *pppoe set transport theylogin* on page 440
ppp <channel> tunnel	Replaced by CLI command *pptp attach* on page 505

Command	CLI Equivalent
ppp <channel> welogin	Replaced by CLI command *pppoe set transport welogin* on page 445
ppp bcp	Replaced by CLI command *pppoe set transport headers llc* on page 414, using the *hdlc* option
ppp interface - localip	Replaced by CLI command *pppoe set transport localip* on page 422
ppp interface - stats	Replaced by CLI command *pppoe show transport* on page 447
ppp user	Blacklisted command, see *user* on page 876
ppp version	Replaced by CLI command *pppoe show transport* on page 447

22.2 pppoe add transport dialout pvc

22.2.1 Syntax

```
pppoe add transport <name> dialout pvc <interface> <port> <vpi> <vci>
[accessconcentrator <concentrator>] [servicename <servicename>]
```

22.2.2 Description

This command creates a PPPoE transport that performs dialout over a PVC (Permanent Virtual Circuit). It allows you to specify the following PVC information:

- the PPP interface to the channel that the PVC will use
- the ATM port that will transport data
- VPI (Virtual Path Identifier)
- VCI (Virtual Circuit Identifier)
- access concentrator (optional)
- service name (optional)

The port/VPI/VCI combination must be unique for each transport.

22.2.3 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	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 PPPoE data. A single interface can be used by multiple channels. The interface value can be any positive integer.	N/A
port	The GlobespanVirata system port that is used to transport ATM data.	N/A
vpi	A field in the ATM header. The VPI is used to identify the virtual path that a circuit belongs to. The VPI can be any value between 0 and 4095.	N/A

Option	Description	Default value
vci	Part of the ATM header. The VCI is a tag that identifies which channel a cell will travel over.The VCI can be any value between 1 and 65535.	N/A
concentrator	A PPPoE tag that identifies a remote access concentrator (or PPPoE server). PPPoE will only connect to the named access concentrator. If no concentrator tag is set, PPPoE connects to the first access concentrator that responds. The tag name/number is determined by your ISP.	N/A
service name	A PPPoE tag that identifies a specific service that is acceptable to the PPPoE client. If set, the PPPoE transport will connect to the first access concentrator it finds that uses this service. If an access concentrator is also set, the PPPoE transport will connect to the specified service on the named concentrator. The service name is determined by your ISP.	N/A

22.2.4 Example

```
prompt> pppoe add transport pppoe1 dialout pvc 1 a1 0 800
accessconcentrator server32 servicename mercury
```

22.2.5 See also

pppoe list transports on page 395

To list ATM ports, see the usable console command BUN *list ports* on page 762.

22.3 pppoe add transport dialout eth

22.3.1 Syntax

```
pppoe add transport <name> dialout eth <interface> <port>
[accessconcentrator <concentrator>] [servicename <servicename>]
```

22.3.2 Description

This command creates a PPPoE transport that performs dialout over Ethernet. It allows you to specify the following Ethernet information:

- the PPP interface to the channel that Ethernet will use
- the Ethernet port that will transport data
- access concentrator (optional)
- service name (optional)

22.3.3 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	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 PPPoE data. A single interface can be used by multiple channels. The interface value can be any positive integer.	N/A
port	The GlobespanVirata system port that is used to transport Ethernet data. To display Ethernet ports, use the *ethernet list ports* command.	N/A

Option	Description	Default value
concentrator	A PPPoE tag that identifies a remote access concentrator (or PPPoE server). PPPoE will only connect to the named access concentrator. If no concentrator tag is set, PPPoE connects to the first access concentrator that responds. The tag name/number is determined by your ISP.	N/A
service name	A PPPoE tag that identifies a specific service that is acceptable to the PPPoE client. If set, the PPPoE transport will connect to the first access concentrator it finds that uses this service. If an access concentrator is also set, the PPPoE transport will connect to the specified service on the named concentrator. The service name is determined by your ISP.	N/A

22.3.4 Example

```
prompt> pppoe add transport pppoe1 dialout eth 1 ethernet
accessconcentrator server40 servicename virginia
```

22.3.5 See also

pppoe list transports on page 395

ethernet list ports on page 169

22.4 pppoe clear transports

22.4.1 Syntax

```
pppoe clear transports
```

22.4.2 Description

This command deletes all PPPoE transports that were created using the *pppoe add transport* commands.

22.4.3 Example

```
prompt> pppoe clear transports
```

22.4.4 See also

pppoe delete transport on page 394

22.5 pppoe delete transport

22.5.1 Syntax

```
pppoe delete transport {<name>|<number>}
```

22.5.2 Description

This command deletes a single PPPoE transport.

22.5.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A

22.5.4 Example

```
prompt> pppoe delete transport pppoe1
```

22.5.5 See also

pppoe list transports on page 395

22.6 pppoe list transports

22.6.1 Syntax

```
pppoe list transports
```

22.6.2 Description

This command lists PPPoE transports that have been created using the *pppoe add transport* command. It displays the following information about the transports:

- transport identification number
- transport name
- Name of port used
- Virtual Circuit Identifier (VCI) used (PVC transports only)
- Virtual Path Identifier (VPI) used (PVC transports only)

22.6.3 Example

```
prompt> pppoe list transports
PPPoE transports:

 ID  |    Name    | Port      | Vci       | Vpi
-----|------------|-----------|-----------|-----------
   1 | p3         | realtek   | N/A       | N/A
   2 | p2         | a1        | 800       | 0
   3 | p1         | ethernet0 | N/A       | N/A
-------------------------------------------------------
```

22.6.4 See also

pppoe show transport on page 447

22.7 pppoe set transport accessconcentrator

22.7.1 Syntax

```
pppoe set transport {<name>|<number>} accessconcentrator <concentrator>
```

22.7.2 Description

This command specifies the access concentrator that you want PPPoE to connect to.

You can also specify a service name using the *set transport servicename* command so that PPPoE will only accept a specific service via a specific access concentrator.

22.7.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
concentrator	A PPPoE tag that identifies a remote access concentrator (or PPPoE server). PPPoE will only connect to the named access concentrator. If no concentrator tag is set, PPPoE connects to the first access concentrator that responds. The tag name/number is determined by your ISP.	Empty string

22.7.4 Example

```
prompt> pppoe set transport pppoe1 accessconcentrator server5
```

22.7.5 See also

pppoe list transports on page 395

pppoe set transport servicename on page 435

pppoe show transport on page 447

For more information on PPPoE and access concentrators, see RFC2516; *http://www.ietf.org/rfc/rfc2516.txt*.

22.8 pppoe set transport autoconnect

22.8.1 Syntax

```
pppoe set transport {<name>|<number>} autoconnect {enabled|disabled}
```

22.8.2 Description

This command allows you to enable/disable the PPPoE autoconnect function. If enabled, PPPoE automatically connects to TCP/IP whenever a user requests TCP/IP packets from a public destination.

22.8.3 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
name	A name that identifies an existing PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
enabled	Enables PPPoE autoconnect.	disabled
disabled	Disables PPPoE autoconnect.	

22.8.4 Example

```
prompt> pppoe set transport pppoe1 autoconnect enabled
```

22.8.5 See also

pppoe list transports on page 395

22.9 pppoe set transport bt

22.9.1 Syntax

```
pppoe set transport {<name>|<number>} bt <burst tolerance>
```

22.9.2 Description

This command sets the burst tolerance (bt) for an existing PPPoE transport. This command is only valid if you set VBR or VBR RT as the QoS Class using the *pppoe set transport qosclass* command.

22.9.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
burst tolerance	Controls the duration of traffic bursts on VBR (Variable Bit Rate) and VBR RT (VBR Real Time) channels. This value overrides an existing MBS value (if set). The BT can be any value between 0 and 100.	0

22.9.4 Example

```
prompt> pppoe set transport pppoe1 bt 5
```

22.9.5 See also

pppoe set transport mbs on page 423

22.10 pppoe set transport createroute

22.10.1 Syntax

```
pppoe set transport {<name>|<number>} createroute {enabled|disabled}
```

22.10.2 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 *pppoe set transport specificroute* command.

To display the route, use the *ip list routes* command. The route is removed when the PPP link is disconnected.

22.10.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe 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.	

22.10.4 Example

```
prompt> pppoe set transport pppoe1 createroute disabled
```

22.10.5 See also

pppoe show transport on page 447

pppoe set transport specificroute on page 437

ip list routes on page 650

22.11 pppoe set transport discoverdns primary

22.11.1 Syntax

```
pppoe set transport {<name>|<number>} discoverdns primary
{enabled|disabled}
```

22.11.2 Description

Note - You must enable one of the *pppoe set transport givedns* commands in order for this command setting to work. See *pppoe set transport givedns client enabled|disabled* on page 408, or *pppoe set transport givedns relay enabled|disabled* on page 410

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 *pppoe set transport givedns* commands is also enabled.

22.11.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe 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.	

22.11.4 Example

```
prompt> pppoe set transport pppoe3 discoverdns primary
enabled
```

22.11.5 See also

pppoe set transport discoverdns secondary on page 404

pppoe set transport givedns client enabled|disabled on page 408

pppoe set transport givedns relay enabled|disabled on page 410

pppoe set transport remotedns on page 430

22.12 pppoe set transport discoverdns secondary

22.12.1 Syntax

```
pppoe set transport {<name>|<number>} discoverdns
secondary {enabled|disabled}
```

22.12.2 Description

Note - You must enable one of the *pppoe set transport givedns* commands in order for this command setting to work. See *pppoe set transport givedns client enabled|disabled* on page 408, or *pppoe set transport givedns relay enabled|disabled* on page 410

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 *pppoe set transport givedns* commands is also enabled.

22.12.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
enabled	A secondary DNS server IP address is requested.	enabled
disabled	A secondary DNS server IP address is not requested.	

22.12.4 Example

```
prompt> pppoe set transport pppoe3 discoverdns secondary
enabled
```

22.12.5 See also

pppoe set transport discoverdns primary on page 402

pppoe set transport givedns client enabled|disabled on page 408

pppoe set transport givedns relay enabled|disabled on page 410

pppoe set transport remotedns on page 430

22.13 pppoe set transport eth

22.13.1 Syntax

```
pppoe set transport {<name>|<number>} eth <port>
```

22.13.2 Description

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

22.13.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
port	The GlobespanVirata system port that is used to transport Ethernet data. To display Ethernet ports, use the *ethernet list ports* command.	N/A

22.13.4 Example

```
prompt> pppoe set transport pppoe3 eth ethernet0
```

22.13.5 See also

pppoe list transports on page 395

ethernet list ports on page 169

22.14 pppoe set transport enabled|disabled

22.14.1 Syntax

```
pppoe set transport {<name>|<number>} {enabled|disabled}
```

22.14.2 Description

This command explicitly enables/disables a PPPoE 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.

22.14.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
enabled	Enables a PPPoE transport.	disabled
disabled	Disables a PPPoE transport.	

22.14.4 Example

```
prompt> pppoe set transport pppoe1 enabled
```

22.14.5 See also

pppoe list transports on page 395

22.15 pppoe set transport givedns client enabled|disabled

22.15.1 Syntax

```
pppoe set transport {<name>|<number>} givedns client {enabled | disabled}
```

22.15.2 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.

22.15.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
enabled	IPCP can request a DNS server IP address and then give the address to DNS client.	disabled
disabled	IPCP cannot request a DNS server IP address and then give the address to DNS client.	

22.15.4 Example

```
prompt> pppoe set transport pppoe1 givedns client enabled
```

22.15.5 See also

pppoe set transport givedns relay enabled|disabled on page 410

pppoe set transport remotedns on page 430

pppoe set transport discoverdns primary on page 402

pppoe set transport discoverdns secondary on page 404

For more information on DNS client, see *ATMOS DNS Client Functional Specification: DO-008322-PS*.

For information on including processes in your image build, see the Software User's Guide for the GlobespanVirata system that you are using.

For information on DNS implementation and specification, see *http://www.ietf.org/rfc/rfc1035.txt*.

22.16 pppoe set transport givedns relay enabled|disabled

22.16.1 See also

22.16.2 Syntax

```
pppoe set transport {<name>|<number>} givedns relay {enabled | disabled}
```

22.16.3 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.

22.16.4 Options

The following table gives 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
enabled	IPCP can request a DNS server IP address and then give the address to DNS relay.	disabled
disabled	IPCP cannot request a DNS server IP address and then give the address to DNS relay.	

22.16.5 Example

```
prompt> pppoe set transport pppoe1 givedns relay enabled
```

22.16.6 See also

pppoe set transport givedns client enabled|disabled on page 408

pppoe set transport discoverdns primary on page 402

pppoe set transport discoverdns secondary on page 404

pppoe set transport remotedns on page 430

DNS Relay CLI commands on page 157

For more information on DNS relay, see *GlobespanVirata DNS Relay Functional Specification: DO-007692-PS*.

For information on including processes in your image build, see the Software User's Guide for the GlobespanVirata system that you are using.

For information on DNS implementation and specification, see *http://www.ietf.org/rfc/rfc1035.txt*.

22.17 pppoe set transport headers hdlc

22.17.1 Syntax

```
pppoe set transport {<name>|<number>} headers hdlc {enabled|disabled}
```

22.17.2 Description

This command allows you to enable/disable whether your system can transmit and receive packets containing HDLC headers. If you want LLC packets to be transmitted and received instead of/as well as HDLC packets, use the *pppoe set transport headers llc enabled* command.

When both HDLC and LLC headers are disabled, the default encapsulation method is VC multiplexed (VC Mux). PPP determines which format to use to transmit/receive packets by 'learning' the format information from incoming packet headers.

22.17.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
enabled	Packets that have HDLC headers can be transmitted/received.	disabled (if LLC headers are disabled too, the default value is VC Mux)
disabled	Packets that have HDLC headers can not be transmitted/received.	

22.17.4 Example

```
prompt> pppoe set transport pppoe1 headers hdlc enabled
```

22.17.5 See also

pppoe list transports on page 395

pppoe show transport on page 447

pppoe set transport headers llc on page 414

22.18 pppoe set transport headers llc

22.18.1 Syntax

```
pppoe set transport {<name>|<number>} headers llc {enabled|disabled}
```

22.18.2 Description

This command allows you to enable/disable whether your system can transmit and receive packets containing LLC headers. If you want HDLC packets to be transmitted and received instead of/as well as LLC packets, use the *pppoe set transport headers hdlc enabled* command.

When both LLC and HDLC headers are disabled, the default encapsulation method is VC multiplexed (VC Mux). PPP determines which format to use to transmit/receive packets by 'learning' the format information from incoming packet headers.

22.18.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
enabled	Packets that have LLC headers can be transmitted/received.	disabled (if HDLC headers are disabled too, the default value is VC Mux)
disabled	Packets that have LLC headers can not be transmitted/received.	

22.18.4 Example

```
prompt> pppoe set transport pppoe1 headers llc enabled
```

22.18.5 See also

pppoe list transports on page 395

pppoe show transport on page 447

pppoe set transport headers llc on page 414

22.19 pppoe set transport idletimeout

22.19.1 Syntax

```
pppoe set transport {<name>|<number>} idletimeout <idletimeout>
```

22.19.2 Description

This command allows you to set an ‘idle’ time out for your LAN connection. If you are connected to an ISP via PPPoE but fail to send a request for data within a specified time limit, the PPPoE session is disabled.

22.19.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
idletimeout	The length of time (in minutes) that a PPPoE session connected to an ISP can remain idle before the session is disabled. The time can be any value between 0 and 60. A value of 0 means that no idletimeout is set.	0

22.19.4 Example

```
prompt> pppoe set transport pppoe1 idletimeout 20
```

22.19.5 See also

pppoe list transports on page 395

pppoe set transport lcpechoevery on page 418

22.20 pppoe set transport interface

22.20.1 Syntax

```
pppoe set transport {<name>|<number>} interface <interface>
```

22.20.2 Description

This command sets the PPP interface for an existing PPPoE transport.

22.20.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
interface	The PPP interface to a channel that transports PPPoE data. A single interface can be used by multiple channels. The interface value can be any positive integer.	N/A

22.20.4 Example

```
prompt> pppoe set transport pppoe2 interface 4
```

22.20.5 See also

pppoe show transport on page 447

pppoe list transports on page 395

22.21 pppoe set transport lcpechoevery

22.21.1 Syntax

```
pppoe set transport {<name>|<number>} lcpechoevery <interval>
```

22.21.2 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.

22.21.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe 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

22.21.4 Example

```
prompt> pppoe set transport pppoe2 lcpechoevery 0
```

22.21.5 See also

pppoe show transport on page 447

pppoe list transports on page 395

22.22 pppoe set transport lcpmaxconf

22.22.1 Syntax

```
pppoe set transport {<name>|<number>} lcpmaxconf <lcp max configure>
```

22.22.2 Description

This command sets the Link Control Protocol (LCP) maximum configure number for an existing PPPoE transport.

22.22.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe 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

22.22.4 Example

```
prompt> pppoe set transport pppoe1 lcpmaxconf 20
```

22.22.5 See also

pppoe show transport on page 447

pppoe list transports on page 395

22.23 pppoe set transport lcpmaxfail

22.23.1 Syntax

```
pppoe set transport {<name>|<number>} lcpmaxfail <lcp max fail>
```

22.23.2 Description

This command sets the Link Control Protocol (LCP) maximum fail number for an existing PPPoE transport.

22.23.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe 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

22.23.4 Example

```
prompt> pppoe set transport pppoe1 lcpmaxfail 20
```

22.23.5 See also

pppoe show transport on page 447

pppoe list transports on page 395

22.24 pppoe set transport lcpmaxterm

22.24.1 Syntax

```
pppoe set transport {<name>|<number>} lcpmaxterm <lcp max terminate>
```

22.24.2 Description

This command sets the Link Control Protocol (LCP) maximum terminate number for an existing PPPoE transport.

22.24.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
lcp 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

22.24.4 Example

```
prompt> pppoe set transport pppoe1 lcpmaxterm 20
```

22.24.5 See also

pppoe show transport on page 447

pppoe list transports on page 395

22.25 pppoe set transport localip

22.25.1 Syntax

```
pppoe set transport {<name>|<number>} localip <ip-address>
```

22.25.2 Description

This command tells the PPP process the local IP address to be associated with the client-end of an interface. This enables remote users to have dialin access via the channel(s) that the interface is attached to.

22.25.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe 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

22.25.4 Example

```
prompt> pppoe set transport pppoe1 localip 192.168.103.2
```

22.25.5 See also

pppoe show transport on page 447

pppoe list transports on page 395

pppoe set transport remoteip on page 432

22.26 pppoe set transport mbs

22.26.1 Syntax

```
pppoe set transport {<name>|<number>} mbs <max burst size>
```

22.26.2 Description

This command sets the maximum burst size (mbs) for an existing PPPoE transport that performs dialout over PVC.

This command is only valid if you set VBR or VBR RT as the QoS Class using the *pppoe set transport quosclass* command.

22.26.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
maximum burst size	Controls the maximum burst size for VBR (Variable Bit Rate) and VBR RT (VBR Real Time) channels. This value overrides an existing BT value (if set). The MBS can be any value between 0 and 100.	0

22.26.4 Example

```
prompt> pppoe set transport pppoe3 mbs 10
```

22.26.5 See also

pppoe set transport bt on page 399

pppoe set transport qosclass on page 428

22.27 pppoe set transport mcr

22.27.1 Syntax

```
pppoe set transport {<name>|<number>} mcr <min cell rate>
```

22.27.2 Description

This command sets the minimum cell rate for an existing PPPoE transport that performs dialout over PVC.

22.27.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
minimum cell rate	Determines the minimum rate at which ATM cells are allowed to be sent along the PPPoE transport.	0

22.27.4 Example

```
prompt> pppoe set transport pppoe2 mcr 0
```

22.27.5 See also

pppoe set transport pcr on page 426

22.28 pppoe set transport password

22.28.1 Syntax

```
pppoe set transport {<name>|<number>} password <password>
```

22.28.2 Description

This command sets a dialout password on a named transport. The password is required when PPP negotiation takes place and is supplied to the remote PPP server for authentication.

22.28.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe 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 *pppoe show transport* command.	N/A

22.28.4 Example

```
prompt> pppoe set transport pppoe2 password mercury
```

22.28.5 See also

pppoe list transports on page 395

pppoe show transport on page 447

pppoe set transport username on page 442

22.29 pppoe set transport pcr

22.29.1 Syntax

```
pppoe set transport {<name>|<number>} pcr <peak cell rate>
```

22.29.2 Description

This command sets the peak cell rate (pcr) for an existing PPPoE transport that performs dialout over PVC.

22.29.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
peak cell rate	Determines the maximum rate at which ATM cells are allowed to be sent along the PPPoE transport. The PCR can be any value from 0 up to the maximum PortSpeed parameter set when the port was created (using the initbun file in FlashFS or the console command BUN *set port* on page 764).	0

22.29.4 Example

```
prompt> pppoe set transport pppoe2 pcr 50000
```

22.29.5 See also

pppoe set transport mcr on page 424

22.30 pppoe set transport port

22.30.1 Syntax

```
pppoe set transport {<name>|<number>} port <port>
```

22.30.2 Description

This command sets the port that is used to transport PPPoE data.

For PVC transports, the port/VPI/VCI combination must be unique for each transport.

22.30.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
port	The GlobespanVirata system port that is used to transport data.	N/A

22.30.4 Example

```
prompt> pppoe set transport pppoe4 port a1
```

22.30.5 See also

pppoe list transports on page 395

To list ATM ports, see the usable console command BUN *list ports* on page 762.

To list Ethernet ports, see the CLI command *ethernet list ports* on page 169.

22.31 pppoe set transport qosclass

22.31.1 Syntax

```
pppoe set transport {<name>|<number>} qosclass {ubr|cbr|vbr|vbrrt|abr|qfc}
```

22.31.2 Description

This command sets the quality of service class for an existing PPPoE transport that performs dialout over PVC.

22.31.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A

Option	Description	Default value
ubr	Unspecified Bit Rate; non-constant and unpredictable data transport rate. PCR (Peak Cell Rate) is the average and maximum speed of transmission.	UBR
cbr	Constant Bit Rate; constant demand and predictable data transport rate. PCR is the average and maximum speed of transmission.	
vbr	Variable Bit Rate; non-constant but predictable data transport rate that uses Non-Real-Time (NRT). You can specify the PCR, SCR, BT and MBS for VBR traffic.	
vbrrt	Variable Bit Rate Real-Time; non-constant but predictable data transport rate that uses Real-Time (RT). You can specify the PCR, SCR, BT and MBS for VBRRT traffic.	
abr	Available Bit Rate; non-constant and unpredictable data transport rate that provides ATM-layer feedback and flow control.	
qfc	Quantum Flow Control; ATM flow control protocol that supports ABR.	

22.31.4 Example

```
prompt> pppoe set transport pppoe3 abr
```

22.31.5 See also

pppoe show transport on page 447

pppoe set transport qosclass on page 428

22.32 pppoe set transport remotedns

22.32.1 Syntax

```
pppoe set transport {<name>|<number>} remotedns
<ipaddress> [<ipaddress2>]
```

22.32.2 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*.

22.32.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe 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)

22.32.4 Examples

Example One - setting a primary address

```
prompt> pppoe set transport pppoe1 remotedns 192.168.102.3
```

Example Two - setting primary and secondary addresses

To set primary and secondary addresses, use this command syntax:

```
prompt> pppoe set transport pppoe1 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> pppoe set transport pppoe1 remotedns 192.168.102.3
0.0.0.0
```

22.32.5 See also

pppoe set transport givedns client enabled|disabled on page 408

pppoe set transport givedns relay enabled|disabled on page 410

pppoe set transport discoverdns primary on page 402

pppoe set transport discoverdns secondary on page 404

22.33 pppoe set transport remoteip

22.33.1 Syntax

```
pppoe set transport {<name>|<number>} remoteip <ip-address>
```

22.33.2 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.

22.33.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
ip-address	The IP address of the local 'server-end' of an interface displayed in the following format: 111.222.254.4	0.0.0.0

22.33.4 Example

```
prompt> pppoe set transport pppoe1 remoteip 192.168.103.2
```

22.33.5 See also

pppoe show transport on page 447

pppoe list transports on page 395

pppoe set transport localip on page 422

22.34 pppoe set transport routemask

22.34.1 Syntax

```
pppoe set transport {<name>|<number>} routemask <mask>
```

22.34.2 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.

22.34.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe 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

22.34.4 Example

```
prompt> pppoe set transport pppoe1 routemask 0.0.0.0
```

22.34.5 See also

pppoe show transport on page 447

pppoe list transports on page 395

22.35 pppoe set transport scr

22.35.1 Syntax

```
pppoe set transport {<name>|<number>} scr <sustainable cell rate>
```

22.35.2 Description

This command sets the sustainable cell rate for an existing PPPoE transport that performs dialout over PVC. This command is only valid if you set VBR or VBR RT as the QoS Class using the *pppoe set transport qosclass* command.

22.35.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
sustainable cell rate	Sustainable Cell Rate; the average cell rate for a VBR or VBR RT connection. The SCR can be any positive value that is less than the PortSpeed and the PCR for the channel.	0

22.35.4 Example

```
prompt> pppoe set transport pppoe2 scr 25000
```

22.35.5 See also

pppoe set transport qosclass on page 428

22.36 pppoe set transport servicename

```
pppoe set transport {<name>|<number>} servicename <servicename>
```

22.36.1 Description

This command specifies the service name that is acceptable to the PPPoE client.

You can also set the access concentrator using the *set transport accessconcentrator* command so that PPPoE will only accept a specific service via a specific access concentrator.

22.36.2 Options

The following table gives 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
service name	A PPPoE tag that identifies a specific service that is acceptable to the PPPoE client. If set, the PPPoE transport will connect to the first access concentrator it finds that uses this service. If an access concentrator is also set, the PPPoE transport will connect to the specified service on the named concentrator. The service name is determined by your ISP.	Empty string

22.36.3 Example

```
prompt> pppoe set transport pppoe1 servicename jupiter
```

22.36.4 See also

pppoe list transports on page 395

pppoe set transport accessconcentrator on page 396

pppoe show transport on page 447

For more information on PPPoE and service names, see RFC2516; *http://www.ietf.org/rfc/rfc2516.txt*.

22.37 pppoe set transport specificroute

22.37.1 Syntax

```
pppoe set transport {<name>|<number>} specificroute {enabled | disabled}
```

22.37.2 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 *pppoe 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 *pppoe set transport createroute* command is set to *disabled*.

22.37.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe 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.	

22.37.4 Example

```
prompt> pppoe set transport pppoe1 specificroute disabled
```

22.37.5 See also

pppoe set transport routemask on page 433

pppoe set transport createroute on page 400

pppoe list transports on page 395

22.38 pppoe set transport subnetmask

22.38.1 Syntax

```
pppoe set transport {<name>|<number>} subnetmask <mask>
```

22.38.2 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.

22.38.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
mask	The subnet mask used by the local IP interface connected to the PPP transport.	0.0.0.0

22.38.4 Example

```
prompt> pppoe set transport pppoe1 subnetmask 255.255.255.0
```

22.38.5 See also

pppoe list transports on page 395

22.39 pppoe set transport theylogin

22.39.1 Syntax

```
pppoe set transport {<name>|<number>} theylogin {none|pap|chap}
```

22.39.2 Description

This command sets the authentication method that remote PPP clients must use to dialin to the server. 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.

22.39.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe 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.	

22.39.4 Example

```
prompt> pppoe set transport pppoe2 theylogin pap
```

22.39.5 See also

pppoe list transports on page 395

pppoe show transport on page 447

system add user on page 591

system set user maydialin on page 617

22.40 pppoe set transport username

22.40.1 Syntax

```
pppoe set transport {<name>|<number>} username <username>
```

22.40.2 Description

This command sets a (dialout) username on a named transport. The username is required when PPP negotiation takes place and is supplied to the remote PPP server for authentication.

22.40.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe 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 digits. To display the username, use the *pppoe show transport* command.	N/A

22.40.4 Example

```
prompt> pppoe set transport pppoe2 username jsmith
```

22.40.5 See also

pppoe set transport password on page 425

22.41 pppoe set transport vci

22.41.1 Syntax

```
pppoe set transport {<name>|<number>} vci <vci>
```

22.41.2 Description

This command sets the Virtual Circuit Identifier for an existing PPPoE transport that performs dialout over PVC.

22.41.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
vci	Part of the ATM header. The VCI is a tag that identifies which channel a cell will travel over. The VCI can be any value between 1 and 65535.	N/A

22.41.4 Example

```
prompt> pppoe set transport pppoe4 vci 800
```

22.41.5 See also

pppoe list transports on page 395

pppoe show transport on page 447

pppoe set transport vpi on page 444

22.42 pppoe set transport vpi

22.42.1 Syntax

```
pppoe set transport {<name>|<number>} vpi <vpi>
```

22.42.2 Description

This command sets the Virtual Path Identifier for an existing PPPoE transport that performs dialout over PVC.

22.42.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
vpi	A field in the ATM header. The VPI is used to identify the virtual path that a circuit belongs to. The VPI can be any value between 0 and 4095.	N/A

22.42.4 Example

```
prompt> pppoe set transport pppoe3 vpi 0
```

22.42.5 See also

pppoe list transports on page 395

pppoe show transport on page 447

pppoe set transport vci on page 443

22.43 pppoe set transport welogin

22.43.1 Syntax

```
pppoe set transport {<name>|<number>} welogin {none|auto|pap|chap}
```

22.43.2 Description

This command sets the authentication protocol used to connect to external PPP servers (dialout).

22.43.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
none	No authentication method is used.	None
auto	The authentication protocol used by the remote PPP server is discovered and used.	
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.	

22.43.4 Example

```
prompt> pppoe set transport pppoe2 theylogin pap
```

22.43.5 See also

pppoe set transport theylogin on page 440

pppoe show transport on page 447

pppoe list transports on page 395

22.44 pppoe show transport

22.44.1 Syntax

```
pppoe show transport {<name>|<number>}
```

22.44.2 Description

This command displays the following information about an existing PPPoE transport:

- Description
- Interface number
- Server - dialin status
- Headers - the data format that the transport can accept or receive
- SVC status (false)

- Local IP address
- Subnet mask
- Remote IP address
- Remote DNS
- Propagate DNS to client (true or false)
- Propagate DNS to relay (true or false)

- Create route (true or false)
- Specific route (true or false)
- Route netmask

- Dialout Username
- Dialout Password
- Dialout Authentication method
- Dialin Authentication method

- LCP Max Configure
- LCP Max Failure
- LCP Max Terminate

- LCP Echo Every

- Autoconnect status (true or false)
- User Idle Timeout setting (in minutes)

- Access concentrator
- Service name

- Port name
- VPI (PVC transport only)
- VCI (PVC transport only)
- Quality of Service (QoS) class (PVC transport only)
- Peak cell rate (PVC transport only)
- Burst tolerance (PVC transport only)
- Sustainable Cell Rate (SCR) (PVC transport only)
- Maximum burst size (MBS) (PVC transport only)
- Maximum Cell Rate (MCR) (PVC transport only)

22.44.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A

22.44.4 Example

```
prompt> pppoe show transport pppoe2
PPP Transport: pppoe2
```

```
      Description: pppoe2
     Interface ID: 1                 Server: false
          Headers: learn             SVC: false
         Local IP: 0.0.0.0
      Subnet mask: 0.0.0.0
        Remote IP: 0.0.0.0
       Remote DNS: 0.0.0.0
Propogate DNS to client: true  To relay: true

     Create route: true
   Specific route: false
    Route netmask: 0.0.0.0

Dialout username:
Dialout password:
    Dialout auth.: none
     Dialin auth.: none

   LCP Max. Conf.: 10
LCP Max. Failure: 5
LCP Max Terminate: 2
   LCP Echo Every: 10

      Autoconnect: true
User Idle Timeout: 30

     Access Conc.:
     Service name: y

       Port: a1          VPI: 0           VCI: 300
  QOS class: UBR
          Peak cell rate: 59111     Burst tolerance: 0
    Sustainable cell rate: 0        Max. burst size: 0
    Sustainable cell rate: 0        Max. burst size: 0
```

22.44.5 See also

pppoe list transports on page 395

23.PPPoH CLI commands

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

23.1 Summary

23.1.1 PPPoH CLI commands

The table below lists the *PPPoH* commands provided by the CLI:

Command	Description/Console command
pppoh add transport dialin	*pppoh add transport dialin* on page 456
pppoh add transport dialout	*pppoh add transport dialout* on page 457
pppoh clear transports	*pppoh clear transports* on page 458
pppoh delete transport	*pppoh delete transport* on page 459
pppoh list transports	*pppoh list transports* on page 460
pppoh set transport createroute	*pppoh set transport createroute* on page 461
pppoh set transport dialin	*pppoh set transport dialin* on page 463
pppoh set transport dialout	*pppoh set transport dialout* on page 464
pppoh set transport discoverdns primary	*pppoh set transport discoverdns primary* on page 465
pppoh set transport discoverdns secondary	*pppoh set transport discoverdns secondary* on page 467
pppoh set transport enabled/disabled	*pppoh set transport enabled|disabled* on page 469
pppoh set transport givedns client enabled/disabled	*pppoh set transport givedns client enabled|disabled* on page 470
pppoh set transport givedns relay enabled/disabled	*pppoh set transport givedns relay enabled|disabled* on page 472
pppoh set transport headers hdlc	*pppoh set transport headers hdlc* on page 474
pppoh set transport headers llc	*pppoh set transport headers llc* on page 476

Command	Description/Console command
pppoh set transport interface	*pppoh set transport interface* on page 477
pppoh set transport lcpechoevery	*pppoh set transport lcpechoevery* on page 478
pppoh set transport lcpmaxconf	*pppoh set transport lcpmaxconf* on page 479
pppoh set transport lcpmaxfail	*pppoh set transport lcpmaxfail* on page 480
pppoh set transport lcpmaxterm	*pppoh set transport lcpmaxterm* on page 481
pppoh set transport localip	*pppoh set transport localip* on page 482
pppoh set transport password	*pppoh set transport password* on page 483
pppoh set transport port	*pppoh set transport port* on page 484
pppoh set transport remotedns	*pppoh set transport remotedns* on page 485
pppoh set transport remoteip	*pppoh set transport remoteip* on page 487
pppoh set transport routemask	*pppoh set transport routemask* on page 488
pppoh set transport specificroute	*pppoh set transport specificroute* on page 489
pppoh set transport subnetmask	*pppoh set transport subnetmask* on page 491
pppoh set transport theylogin	*pppoh set transport theylogin* on page 492
pppoh set transport username	*pppoh set transport username* on page 494
pppoh set transport welogin	*pppoh set transport welogin* on page 495
pppoh show transport	*pppoh show transport* on page 497

23.1.2 PPP Console commands

The table below lists the *ppp* **console** commands and, if available, their equivalent CLI command:

Command	CLI Equivalent
ppp <channel> clear	Blacklisted command, see *<channel> clear* on page 850
ppp <channel> disable	Blacklisted command, see *<channel> disable* on page 851
ppp <channel> discard	Usable command, see *<channel> discard* on page 852
ppp <channel> echo	Usable command, see *<channel> echo* on page 853
ppp <channel> echo every	Usable command, see *<channel> echo every* on page 854
ppp <channel> enable	Blacklisted command, see *<channel> enable* on page 855
ppp <channel> event	Usable command, see *<channel> event* on page 856
ppp <channel> hdlc	Replaced by CLI command *pppoh set transport headers hdlc* on page 474, using the *hdlc* option.
ppp <channel> info	Replaced by CLI command *pppoh show transport* on page 497.
ppp <channel> interface	Replaced by CLI command *pppoh set transport interface* on page 477
ppp <channel> lcpmaxconfigure	Replaced by CLI command *pppoh set transport lcpmaxconf* on page 479
ppp <channel> lcpmaxfailure	Replaced by CLI command *pppoh set transport lcpmaxfail* on page 480
ppp <channel> lcpmaxterminate	Blacklisted command, see *<channel> lcpmaxterminate* on page 862
ppp <channel> llc	Replaced by CLI command *pppoh set transport headers llc* on page 476, using the *llc* option.

Command	CLI Equivalent
ppp <channel> pvc	Replaced by CLI command *pppoh set transport dialin* on page 463 and *pppoh set transport dialout* on page 464
ppp <channel> remoteip	Replaced by CLI command *pppoh set transport remoteip* on page 487.
ppp <channel> svc	Replaced by CLI command and *pppoh set transport enabled|disabled* on page 469
ppp <channel> theylogin	Replaced by CLI command *pppoh set transport theylogin* on page 492
ppp <channel> tunnel	Blacklisted command, see *<channel> tunnel <n> <tunnel protocol> <dial direction>* on page 871
ppp <channel> welogin	Replaced by CLI command *pppoh set transport welogin* on page 495
ppp interface - localip	Replaced by CLI command *pppoh set transport localip* on page 482
ppp interface - stats	Replaced by CLI command *pppoh show transport* on page 497
ppp user	Blacklisted command, see *user* on page 876
ppp version	Blacklisted command, see *version* on page 877

23.2 pppoh add transport dialin

23.2.1 Syntax

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

23.2.2 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

23.2.3 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	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 GlobespanVirata system port that is used to transport HDLC data.	N/A

23.2.4 Example

```
prompt> pppoh add transport pppoh1 dialin 1 hdlc
```

23.2.5 See also

pppoh list transports on page 460

23.3 pppoh add transport dialout

23.3.1 Syntax

```
pppoh add transport <name> dialout <interface> <port>
```

23.3.2 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

23.3.3 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	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 GlobespanVirata system port that is used to transport HDLC data.	N/A

23.3.4 Example

```
prompt> pppoh add transport pppoh1 dialout 1 hdlc
```

23.3.5 See also

pppoh list transports on page 460

23.4 pppoh clear transports

23.4.1 Syntax

```
pppoh clear transports
```

23.4.2 Description

This command deletes all PPPoH transports that were created using the *pppoh add transport* commands.

23.4.3 Example

```
prompt> pppoh clear transports
```

23.4.4 See also

pppoh delete transport on page 459

23.5 pppoh delete transport

23.5.1 Syntax

```
pppoh delete transport {<name>|<number>}
```

23.5.2 Description

This command deletes a single PPPoH transport.

23.5.3 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

23.5.4 Example

```
prompt> pppoh delete transport pppoh1
```

23.5.5 See also

pppoh list transports on page 460

23.6 pppoh list transports

23.6.1 Syntax

```
pppoh list transports
```

23.6.2 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

23.6.3 Example

```
prompt> pppoh list transports
PPPOH transports:

 ID   |    Name
------|------------
   1  | p2
   2  | p1
------------------
```

23.6.4 See also

pppoh show transport on page 497

23.7 pppoh set transport createroute

23.7.1 Syntax

```
pppoh set transport {<name>|<number>} createroute {enabled|disabled}
```

23.7.2 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 *pppoh set transport specificroute* command.

To display the route, use the *ip list routes* command. The route is removed when the PPP link is disconnected.

23.7.3 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 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.	

23.7.4 Example

```
prompt> pppoh set transport pppoh1 createroute disabled
```

23.7.5 See also

pppoh show transport on page 497

pppoh set transport specificroute on page 489

ip list routes on page 650

23.8 pppoh set transport dialin

23.8.1 Syntax

```
pppoh set transport {<name>|<number>} dialin
```

23.8.2 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.

23.8.3 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
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

23.8.4 Example

```
prompt> pppoh set transport pppoh2 dialin
```

23.8.5 See also

pppoh list transports on page 460

pppoh set transport dialout on page 464

23.9 pppoh set transport dialout

23.9.1 Syntax

```
pppoh set transport {<name>|<number>} dialout
```

23.9.2 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.

23.9.3 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 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

23.9.4 Example

```
prompt> pppoh set transport pppoh2 dialout
```

23.9.5 See also

pppoh list transports on page 460

pppoh set transport dialin on page 463

23.10 pppoh set transport discoverdns primary

23.10.1 Syntax

```
pppoh set transport {<name>|<number>} discoverdns primary
{enabled|disabled}
```

23.10.2 Description

Note - You must enable one of the *pppoh set transport givedns* commands in order for this command setting to work. See *pppoh set transport givedns client enabled|disabled* on page 470, or *pppoh set transport givedns relay enabled|disabled* on page 472

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.

23.10.3 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 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
enabled	A primary DNS server IP address is requested.	enabled
disabled	A primary DNS server IP address is not requested.	

23.10.4 Example

```
prompt> pppoh set transport pppoh3 discoverdns primary
enabled
```

23.10.5 See also

pppoh set transport discoverdns secondary on page 467

pppoh set transport givedns client enabled|disabled on page 470

pppoh set transport givedns relay enabled|disabled on page 472

pppoh set transport remotedns on page 485

23.11 pppoh set transport discoverdns secondary

23.11.1 Syntax

```
pppoh set transport {<name>|<number>} discoverdns
secondary {enabled|disabled}
```

23.11.2 Description

Note - You must enable one of the *pppoh set transport givedns* commands in order for this command setting to work. See *pppoh set transport givedns client enabled|disabled* on page 470, or *pppoh set transport givedns relay enabled|disabled* on page 472

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.

23.11.3 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 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
enabled	A secondary DNS server IP address is requested.	enabled
disabled	A secondary DNS server IP address is not requested.	

23.11.4 Example

```
prompt> pppoh set transport pppoh3 discoverdns secondary
enabled
```

23.11.5 See also

pppoh set transport discoverdns primary on page 465

pppoh set transport givedns client enabled|disabled on page 470

pppoh set transport givedns relay enabled|disabled on page 472

pppoh set transport remotedns on page 485

23.12 pppoh set transport enabled|disabled

23.12.1 Syntax

```
pppoh set transport {<name>|<number>} {enabled|disabled}
```

23.12.2 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.

23.12.3 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.	disabled
disabled	Disables a PPPoH transport.	

23.12.4 Example

```
prompt> pppoh set transport pppoh1 enabled
```

23.12.5 See also

pppoh list transports on page 460

23.13 pppoh set transport givedns client enabled|disabled

23.13.1 Syntax

```
pppoh set transport {<name>|<number>} givedns client {enabled | disabled}
```

23.13.2 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.

23.13.3 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 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
enabled	IPCP can request a DNS server IP address and then give the address to DNS client.	disabled
disabled	IPCP cannot request a DNS server IP address and then give the address to DNS client.	

23.13.4 Example

```
prompt> pppoh set transport pppoh1 givedns client enabled
```

23.13.5 See also

pppoh set transport givedns relay enabled|disabled on page 472

pppoh set transport remotedns on page 485

pppoh set transport discoverdns primary on page 465

pppoh set transport discoverdns secondary on page 467

DNS Client CLI commands on page 147

For more information on DNS client, see *ATMOS DNS Client Functional Specification: DO-008322-PS*.

For information on including processes in your image build, see the Software User's Guide for the GlobespanVirata system that you are using.

For information on DNS implementation and specification, see *http://www.ietf.org/rfc/rfc1035.txt*.

23.14 pppoh set transport givedns relay enabled|disabled

23.14.1 Syntax

```
pppoh set transport {<name>|<number>} givedns relay {enabled | disabled}
```

23.14.2 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.

23.14.3 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 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
enabled	IPCP can request a DNS server IP address and then give the address to DNS relay.	disabled
disabled	IPCP cannot request a DNS server IP address and then give the address to DNS relay.	

23.14.4 Example

```
prompt> pppoh set transport pppoh1 givedns relay enabled
```

23.14.5 See also

pppoh set transport givedns client enabled|disabled on page 470

pppoh set transport discoverdns primary on page 465

pppoh set transport discoverdns secondary on page 467

pppoh set transport remotedns on page 485

DNS Relay CLI commands on page 157

For more information on DNS relay, see *GlobespanVirata DNS Relay Functional Specification: DO-007692-PS*.

For information on including processes in your image build, see the Software User's Guide for the GlobespanVirata system that you are using.

For information on DNS implementation and specification, see *http://www.ietf.org/rfc/rfc1035.txt*.

23.15 pppoh set transport headers hdlc

23.15.1 Syntax

```
pppoh set transport {<name>|<number>} headers hdlc {enabled|disabled}
```

23.15.2 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.

23.15.3 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 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
enabled	Packets that have HDLC headers can be transmitted/received.	enabled
disabled	Packets that have HDLC headers can not be transmitted/received.	

23.15.4 Example

```
prompt> pppoh set transport pppoh1 headers hdlc enabled
```

23.15.5 See also

pppoh list transports on page 460

pppoh show transport on page 497

pppoh set transport headers llc on page 476

23.16 pppoh set transport headers llc

23.16.1 Syntax

```
pppoh set transport {<name>|<number>} headers llc {enabled|disabled}
```

23.16.2 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.

23.16.3 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 PPPoE transport. To display transport names, use the *pppoe list transports* command.	N/A
number	A number that identifies an existing PPPoE transport. To display transport numbers, use the *pppoe list transports* command.	N/A
enabled	Packets that have LLC headers can be transmitted/received.	enabled
disabled	Packets that have LLC headers can not be transmitted/received.	

23.16.4 Example

```
prompt> pppoh set transport pppoh1 headers llc enabled
```

23.16.5 See also

pppoh list transports on page 460

pppoh show transport on page 497

pppoh set transport headers hdlc on page 474

23.17 pppoh set transport interface

23.17.1 Syntax

```
pppoh set transport {<name>|<number>} interface <interface>
```

23.17.2 Description

This command sets the PPP interface for an existing PPPoH transport.

23.17.3 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 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
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

23.17.4 Example

```
prompt> pppoh set transport pppoh2 interface 4
```

23.17.5 See also

pppoh show transport on page 497

pppoh list transports on page 460

23.18 pppoh set transport lcpechoevery

23.18.1 Syntax

```
pppoh set transport {<name>|<number>} lcpechoevery <interval>
```

23.18.2 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.

23.18.3 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 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
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

23.18.4 Example

```
prompt> pppoh set transport pppoh2 lcpechoevery 0
```

23.18.5 See also

pppoh show transport on page 497

pppoh list transports on page 460

23.19 pppoh set transport lcpmaxconf

23.19.1 Syntax

```
pppoh set transport {<name>|<number>} lcpmaxconf <lcp max configure>
```

23.19.2 Description

This command sets the Link Control Protocol (LCP) maximum parameter for an existing PPPoH transport.

23.19.3 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 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
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

23.19.4 Example

```
prompt> pppoh set transport pppoh1 lcpmaxconf 20
```

23.19.5 See also

pppoh show transport on page 497

pppoh list transports on page 460

23.20 pppoh set transport lcpmaxfail

23.20.1 Syntax

```
pppoh set transport {<name>|<number>} lcpmaxfail <lcp max fail>
```

23.20.2 Description

This command sets the Link Control Protocol (LCP) maximum fail parameter for an existing PPPoH transport.

23.20.3 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 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
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

23.20.4 Example

```
prompt> pppoh set transport pppoh1 lcpmaxfail 20
```

23.20.5 See also

pppoh show transport on page 497

pppoh list transports on page 460

23.21 pppoh set transport lcpmaxterm

23.21.1 Syntax

```
pppoh set transport {<name>|<number>} lcpmaxterm <lcp max terminate>
```

23.21.2 Description

This command sets the Link Control Protocol (LCP) maximum terminate parameter for an existing PPPoH transport.

23.21.3 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 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
lcp 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

23.21.4 Example

```
prompt> pppoh set transport pppoh1 lcpmaxterm 20
```

23.21.5 See also

pppoh show transport on page 497

pppoh list transports on page 460

23.22 pppoh set transport localip

23.22.1 Syntax

```
pppoh set transport {<name>|<number>} localip <ip-address>
```

23.22.2 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.

23.22.3 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 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
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

23.22.4 Example

```
prompt> pppoh set transport pppoh1 localip 192.168.103.2
```

23.22.5 See also

pppoh show transport on page 497

pppoh list transports on page 460

pppoh set transport remoteip on page 487

23.23 pppoh set transport password

23.23.1 Syntax

```
pppoh set transport {<name>|<number>} password <password>
```

23.23.2 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.

23.23.3 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 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
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

23.23.4 Example

```
prompt> pppoh set transport pppoh2 password mercury
```

23.23.5 See also

pppoh list transports on page 460

pppoh show transport on page 497

pppoh set transport username on page 494

23.24 pppoh set transport port

23.24.1 Syntax

```
pppoh set transport {<name>|<number>} port <port>
```

23.24.2 Description

This command applies to existing transports. This command sets the port that an existing transport uses to transport PPPoH data.

23.24.3 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 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 transport* command.	N/A
port	The GlobespanVirata system port that is used to transport HDLC data.	N/A

23.24.4 Example

```
prompt> pppoh set transport pppoh4 port hdlc
```

23.24.5 See also

pppoh list transports on page 460

23.25 pppoh set transport remotedns

23.25.1 Syntax

```
pppoh set transport {<name>|<number>} remotedns
<ipaddress> [<ipaddress2>]
```

23.25.2 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*.

23.25.3 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 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
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)

23.25.4 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
```

23.25.5 See also

pppoh set transport givedns client enabled|disabled on page 470

pppoh set transport givedns relay enabled|disabled on page 472

pppoh set transport discoverdns primary on page 465

pppoh set transport discoverdns secondary on page 467

23.26 pppoh set transport remoteip

23.26.1 Syntax

```
pppoh set transport {<name>|<number>} remoteip <ip-address>
```

23.26.2 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.

23.26.3 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 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
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

23.26.4 Example

```
prompt> pppoh set transport pppoh1 remoteip 192.168.103.2
```

23.26.5 See also

pppoh show transport on page 497

pppoh list transports on page 460

pppoh set transport localip on page 482

23.27 pppoh set transport routemask

23.27.1 Syntax

```
pppoh set transport {<name>|<number>} routemask <mask>
```

23.27.2 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.

23.27.3 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 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
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

23.27.4 Example

```
prompt> pppoh set transport pppoh1 routemask 0.0.0.0
```

23.27.5 See also

pppoh show transport on page 497

pppoh list transports on page 460

23.28 pppoh set transport specificroute

23.28.1 Syntax

```
pppoh set transport {<name>|<number>} specificroute {enabled | disabled}
```

23.28.2 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*.

23.28.3 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 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
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.	

23.28.4 Example

```
prompt> pppoh set transport pppoh1 specificroute disabled
```

23.28.5 See also

pppoh set transport routemask on page 488

pppoh set transport createroute on page 461

pppoh list transports on page 460

23.29 pppoh set transport subnetmask

23.29.1 Syntax

```
pppoh set transport {<name>|<number>} subnetmask <mask>
```

23.29.2 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.

23.29.3 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 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
mask	The subnet mask used by the local IP interface connected to the PPP transport.	0.0.0.0

23.29.4 Example

```
prompt> pppoh set transport pppoh1 subnetmask 255.255.255.0
```

23.29.5 See also

pppoh list transports on page 460

23.30 pppoh set transport theylogin

23.30.1 Syntax

```
pppoh set transport {<name>|<number>} theylogin {none|pap|chap}
```

23.30.2 Description

This command sets the authentication method that remote PPP clients must use to dialin to the server. 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.

23.30.3 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 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
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.	

23.30.4 Example

```
prompt> pppoh set transport pppoh2 theylogin pap
```

23.30.5 See also

pppoh list transports on page 460

pppoh show transport on page 497

system add user on page 591

system set user maydialin on page 617

23.31 pppoh set transport username

23.31.1 Syntax

```
pppoh set transport {<name>|<number>} username <username>
```

23.31.2 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.

23.31.3 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 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
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 digits. To display the username, use the *pppoh show transport* command.	N/A

23.31.4 Example

```
prompt> pppoh set transport pppoh2 username jsmith
```

23.31.5 See also

pppoh set transport password on page 483

23.32 pppoh set transport welogin

23.32.1 Syntax

```
pppoh set transport {<name>|<number>} welogin {none|auto|pap|chap}
```

23.32.2 Description

This command sets the authentication protocol used to connect to external PPP servers (dial-out).

23.32.3 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 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
none	No authentication method is used.	None
auto	The authentication protocol used by the remote PPP server is discovered and used.	
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.	

23.32.4 Example

```
prompt> pppoh set transport pppoh2 theylogin pap
```

23.32.5 See also

pppoh set transport theylogin on page 492

pppoh show transport on page 497

pppoh list transports on page 460

23.33 pppoh show transport

23.33.1 Syntax

```
pppoh show transport {<name>|<number>}
```

23.33.2 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

23.33.3 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 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

23.33.4 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
```

23.33.5 See also

pppoh list transports on page 460

24.PPTP CLI commands

This chapter describes the PPTP CLI commands.

24.1 Summary

24.1.1 PPTP CLI commands

The table below lists the CLI commands for manipulating PPTP tunnels:

Command	Reference
pptp add tunnel	*pptp add tunnel* on page 504
pptp attach	*pptp attach* on page 505
pptp clear tunnels	*pptp clear tunnels* on page 507
pptp delete tunnel	*pptp delete tunnel* on page 508
pptp detach	*pptp detach* on page 509
pptp list tunnels	*pptp list tunnels* on page 510
pptp set listening	*pptp set listening* on page 511
pptp set localaddr	*pptp set localaddr* on page 512
pptp set tunnel remoteip	*pptp set tunnel remoteip* on page 513
pptp set tunnel type	*pptp set tunnel type* on page 514
pptp show tunnel	*pptp show tunnel* on page 515
pptp show	*pptp show* on page 516

24.1.2 PPTP Console commands

The table below lists the *pptp* **console** commands and, if available, their CLI equivalent command:

Command	CLI Equivalent
pptp bind	Replaced by CLI command *pptp set listening* on page 511.
pptp <tunnel> connect	Replaced by CLI command *pptp attach* on page 505
pptp <tunnel> create	Replaced by CLI command *pptp add tunnel* on page 504

Command	CLI Equivalent
pptp <tunnel> delete	Replaced by CLI command *pptp delete tunnel* on page 508
pptp <tunnel> disconnect	Replaced by CLI command *pptp detach* on page 509
pptp <tunnel> event	Blacklisted command, see *<tunnel> event* on page 887
pptp <tunnel> info	Blacklisted command, see *<tunnel> info* on page 888
pptp list	Usable command, see *list* on page 889
pptp version	Usable command, see *version* on page 890

24.2 pptp add tunnel

24.2.1 Syntax

```
pptp add tunnel <name>
```

24.2.2 Description

This command creates a PPP tunnel.

24.2.3 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	An arbitrary name that identifies the tunnel. 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

24.2.4 Example

```
prompt> pptp add tunnel tunnel1
```

24.2.5 See also

pptp attach on page 505

pptp list tunnels on page 510

For information on creating and attaching tunnels and PPP transports, see the Software User's Guide for the GlobespanVirata system that you are using.

24.3 pptp attach

24.3.1 Syntax

```
pptp attach {<name>|<number>} <ppp-transport>
```

24.3.2 Description

This command attaches an existing tunnel to an existing PPP transport to allow data to be tunnelled via PPP.

Only one tunnel can be attached to a PPP transport. If you use this command when there is already a tunnel attached to the transport, the previous tunnel is replaced by the new one.

This command implicitly enables the PPP transport being attached.

24.3.3 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 tunnel. To display tunnel names, use the *pptp list tunnels* command.	N/A
number	A number that identifies an existing tunnel. To display tunnel numbers, use the *pptp list tunnels* command.	N/A
ppp-transport	A name that identifies an existing PPP transport. To display transport names, use the *pppoa/pppoe list transports* command.	N/A

24.3.4 Example

```
prompt> pptp attach tunnel1 pppoa1
```

24.3.5 See also

pptp add tunnel on page 504

pptp list tunnels on page 510

pppoa list transports on page 325

pppoe list transports on page 395

24.4 pptp clear tunnels

24.4.1 Syntax

```
pptp clear tunnels
```

24.4.2 Description

This command deletes all tunnels that were created using the *pptp add tunnel* command.

24.4.3 Example

```
prompt> pptp clear tunnels
```

24.4.4 See also

pptp delete tunnel on page 508

24.5 pptp delete tunnel

24.5.1 Syntax

```
pptp delete tunnel {<name>|<number>}
```

24.5.2 Description

This command deletes a single PPTP tunnel.

24.5.3 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 tunnel. To display tunnel names, use the *pptp list tunnels* command.	N/A
number	A number that identifies an existing tunnel. To display tunnel numbers, use the *pptp list tunnels* command.	N/A

24.5.4 Example

```
prompt> pptp delete tunnel tunnel1
```

24.5.5 See also

pptp list tunnels on page 510

24.6 pptp detach

24.6.1 Syntax

```
pptp detach {<name>|<number>}
```

24.6.2 Description

This command detaches a single tunnel that was attached to a PPP transport using the *pptp attach tunnel* command.

24.6.3 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 tunnel. To display tunnel names, use the *pptp list tunnels* command.	N/A
number	A number that identifies an existing tunnel. To display tunnel numbers, use the *pptp list tunnels* command.	N/A

24.6.4 Example

```
prompt> pptp detach tunnel1
```

24.6.5 See also

pptp list tunnels on page 510

24.7 pptp list tunnels

24.7.1 Syntax

```
pptp list tunnels
```

24.7.2 Description

This command lists all PPTP tunnels that have been created using the *pptp add tunnel* command. It displays the following information about PPTP tunnels:

- tunnel ID number
- tunnel name
- type (dialin or dialout)
- remote IP address (if applicable)
- whether the tunnel is connected to a named PPP transport
- name of attached PPP transport (if applicable)

24.7.3 Example

```
prompt> pptp list tunnels
PPTP Tunnels:

 ID  |   Name    |    Type    |    Remote IP   | Connected |  Attached
-----|-----------|------------|----------------|-----------|--------------
   1 | tunnel1   | dialin     | 192.168.102.3  | true      | pppoa1
   2 | tunnel2   | dialin     | 0.0.0.0        | false     | Not attached
-------------------------------------------------------------------------
```

24.7.4 See also

pptp show tunnel on page 515

24.8 pptp set listening

24.8.1 Syntax

```
pptp listening {enabled|disabled}
```

24.8.2 Description

This command determines whether PPTP tunnels can accept incoming connections from a remote PPTP Network Server (PNS).

24.8.3 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
enabled	Incoming connections are accepted by PPTP tunnels.	enabled
disabled	Incoming connections are not accepted by PPTP tunnels.	

24.8.4 Example

```
prompt> pptp set listening disabled
```

24.9 pptp set localaddr

24.9.1 Syntax

```
pptp set localaddr <ip address>
```

24.9.2 Description

This command sets the IP address of the local system running the PPTP Access Concentrator (PAC). This is used for incoming and outgoing tunnel connections.

24.9.3 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
ip address	The IP address of the local system running the PAC, displayed in the following format: 192.168.102.3	0.0.0.0

24.9.4 Example

```
prompt> pptp set localaddr 192.168.102.3
```

24.9.5 See also

pptp set tunnel remoteip on page 513

24.10 pptp set tunnel remoteip

24.10.1 Syntax

```
pptp set tunnel {<name>|<number>} remoteip <ip address>
```

24.10.2 Description

This command sets the IP address of the remote end of the tunnel. The remote IP address is for a system running PNS (PPTP Network Server).

24.10.3 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 tunnel. To display tunnel names, use the *pptp list tunnels* command.	N/A
number	A number that identifies an existing tunnel. To display tunnel numbers, use the *pptp list tunnels* command.	N/A
ip address	The IP address of the remote system running the PNS, displayed in the following format: 192.168.102.3	0.0.0.0

24.10.4 Example

```
prompt> pptp set tunnel1 remoteip 192.168.103.2
```

24.10.5 See also

pptp set localaddr on page 512

24.11 pptp set tunnel type

24.11.1 Syntax

```
pptp set tunnel {<name>|<number>} type {<dialin>|<dialout>}
```

24.11.2 Description

This command sets the type of connection (dialin or dialout) that an existing tunnel can use.

24.11.3 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 tunnel. To display tunnel names, use the *pptp list tunnels* command.	N/A
number	A number that identifies an existing tunnel. To display tunnel numbers, use the *pptp list tunnels* command.	N/A
dialin	This allows incoming calls (dialin) to be tunnelled to a specific PNS.	dialin
dialout	This allows PNS-capable machines on a local network to dialout.	

24.11.4 Example

```
prompt> pptp set tunnel tunnel1 type dialout
```

24.11.5 See also

pptp list tunnels on page 510

24.12 pptp show tunnel

24.12.1 Syntax

```
pptp show tunnel {<name>|<number>}
```

24.12.2 Description

This command displays the following information about an existing tunnel:

- name
- connection type (dialin or dialout)
- remote IP address (if set)
- transport connection status

24.12.3 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 tunnel. To display tunnel names, use the *pptp list tunnels* command.	N/A
number	A number that identifies an existing tunnel. To display tunnel numbers, use the *pptp list tunnels* command.	N/A

24.12.4 Example

```
prompt> pptp show tunnel tunnel1
PPTP Tunnel: tunnel1

        Type: dialin
   Remote IP: 192.168.103.2
   Connected: true
```

24.12.5 See also

pptp show on page 516

24.13 pptp show

24.13.1 Syntax

```
pptp show
```

24.13.2 Description

This command displays the following information about your PPTP setup:

- local IP address (if set)
- listening status

24.13.3 Example

```
prompt> pptp show
Global PPTP Configuration:

  Local address: 192.168.102.3
      Listening: true
```

24.13.4 See also

pptp show tunnel on page 515

pptp set localaddr on page 512

pptp set listening on page 511

25.RFC1483 CLI commands

This chapter describes the RFC1483 CLI commands.

25.1 Summary

25.1.1 RFC1483 CLI commands

The table below lists the RFC1483 commands provided by the CLI:

Command	Description/Console command
rfc1483 add transport	*rfc1483 add transport* on page 519
rfc1483 clear transports	*rfc1483 clear transports* on page 521
rfc1483 delete transport	*rfc1483 delete transport* on page 522
rfc1483 list transports	*rfc1483 list transports* on page 523
rfc1483 set transport bt	*rfc1483 set transport bt* on page 524
rfc1483 set transport mbs	*rfc1483 set transport mbs* on page 525
rfc1483 set transport mcr	*rfc1483 set transport mcr* on page 526
rfc1483 set transport mode	*rfc1483 set transport mode* on page 527
rfc1483 set transport pcr	*rfc1483 set transport pcr* on page 528
rfc1483 set transport port	*rfc1483 set transport port* on page 529
rfc1483 set transport qosclass	*rfc1483 set transport qosclass* on page 530
rfc1483 set transport rxvci	*rfc1483 set transport rxvci* on page 532
rfc1483 set transport rxvpi	*rfc1483 set transport rxvpi* on page 533
rfc1483 set transport scr	*rfc1483 set transport scr* on page 534
rfc1483 set transport txvci	*rfc1483 set transport txvci* on page 535
rfc1483 set transport txvpi	*rfc1483 set transport txvpi* on page 536
rfc1483 set transport vci	*rfc1483 set transport vci* on page 537
rfc1483 set transport vpi	*rfc1483 set transport vpi* on page 538
rfc1483 show transport	*rfc1483 show transport* on page 539

25.1.2 RFC1483 Console commands

The above CLI commands replace all previous RFC1483 console commands.

25.2 rfc1483 add transport

25.2.1 Syntax

```
rfc1483 add transport <name> <port> <vpi> <vci>
{llc|vcmux} {bridged|routed}
```

25.2.2 Description

This command creates a named RFC1483 transport and allows you to specify the following:

- the ATM port that will transport RFC1483 data. (ATM ports are initialised in the initbun file in FlashFS, or using the *bun set port* console command.)
- VPI (Virtual Path Identifier)
- VCI (Virtual Circuit Identifier)
- LLC or VcMux encapsulation (optional)
- Bridged or Routed (optional)

The port/VPI/VCI combination must be unique for each transport.

25.2.3 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	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 GlobespanVirata system port that is used to transport ATM data.	N/A
vpi	A field in the ATM header. The VPI is used to identify the virtual path that a circuit belongs to. The VPI can be any value between 0 and 4095.	N/A
vci	Part of the ATM header. The VCI is a tag that identifies which channel a cell will travel over. The VCI can be any value between 1 and 65535.	N/A

Option	Description	Default value
llc	Logical Link Control encapsulation method.	llc
vcmux	VC Multiplexing encapsulation method.	
bridged	Traffic type that is going to be transmitted/received.	bridged
routed	Traffic type that is going to be transmitted/received.	

25.2.4 Example

```
prompt> rfc1483 add transport my1483 myport 0 700 vcmux routed
```

25.2.5 See also

rfc1483 list transports on page 523

Console command BUN *set port* on page 764

To list ATM ports, see the usable console command *list ports* on page 762.

25.3 rfc1483 clear transports

25.3.1 Syntax

```
rfc1483 clear transports
```

25.3.2 Description

This command deletes all RFC1483 transports that were created using the *rfc1483 add transport* command.

25.3.3 Example

```
prompt> rfc1483 clear transports
```

25.3.4 See also

rfc1483 delete transport on page 522

25.4 rfc1483 delete transport

25.4.1 Syntax

```
rfc1483 delete transport {<name>|<number>}
```

25.4.2 Description

This command deletes a single RFC1483 transport.

25.4.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A

25.4.4 Example

```
prompt> rfc1483 delete transport my1483
```

25.4.5 See also

rfc1483 list transports on page 523

25.5 rfc1483 list transports

25.5.1 Syntax

```
rfc1483 list transports
```

25.5.2 Description

This command lists all rfc1483 transports that have been created using the *rfc1483 add transport* command. It displays the following information about the transports:

- transport identification number
- transport name
- name of the ATM port used to transport rfc1483 data
- transmit and receive VCI numbers
- transmit and receive VPI numbers

25.5.3 Example

```
prompt> rfc1483 list transports
RFC1483 transports:

 ID  |    Name    | Port      | TxVci     | RxVci     | TxVpi     | RxVpi
-----|------------|-----------|-----------|-----------|-----------|--------
   1 | my1483     | a1        | 700       | 700       | 0         | 0
-------------------------------------------------------------------------
```

25.5.4 See also

rfc1483 show transport on page 539

25.6 rfc1483 set transport bt

25.6.1 Syntax

```
rfc1483 set transport {<name>|<number>} bt <burst tolerance>
```

25.6.2 Description

This command sets the burst tolerance (bt) that an existing RFC1483 transport uses to transport data over ATM. This command is only valid if you set VBR or VBR RT as the QoS Class using the *rfc1483 set transport qosclass* command.

25.6.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A
burst tolerance	Controls the duration of traffic bursts on VBR (Variable Bit Rate) and VBR RT (VBR Real Time) channels. This value overrides an existing MBS value (if set). The BT can be any value between 0 and 100.	0

25.6.4 Example

```
prompt> rfc1483 set transport my1483 bt 5
```

25.6.5 See also

rfc1483 set transport mbs on page 525

rfc1483 set transport qosclass on page 530

25.7 rfc1483 set transport mbs

25.7.1 Syntax

```
rfc1483 set transport {<name>|<number>} mbs <maximum burst size>
```

25.7.2 Description

This command sets the maximum burst size (mbs) for the RFC1483 transport. This command is only valid if you set VBR or VBR RT as the QoS Class using the *rfc1483 set transport qosclass* command.

25.7.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A
maximum burst size	Controls the maximum burst size for VBR (Variable Bit Rate) and VBR RT (VBR Real Time) channels. This value overrides an existing BT value (if set). The MBS can be any value between 0 and 100.	0

25.7.4 Example

```
prompt> rfc1483 set transport my1483 mbs 10
```

25.7.5 See also

rfc1483 set transport bt on page 524

rfc1483 set transport qosclass on page 530

25.8 rfc1483 set transport mcr

25.8.1 Syntax

```
rfc1483 set transport {<name>|<number>} mcr <minimum cell rate>
```

25.8.2 Description

This command sets the minimum cell rate (mcr) for an existing RFC1483 transport.

25.8.3 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
name	A name that identifies an existing RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A
minimum cell rate	Determines the minimum rate at which ATM cells are allowed to be transported into the ATM network.	0

25.8.4 Example

```
prompt> rfc1483 set transport my1483 mcr 0
```

25.8.5 See also

rfc1483 set transport pcr on page 528

25.9 rfc1483 set transport mode

25.9.1 Syntax

```
rfc1483 set transport {<name>|<number>} mode {llc|vcmux}
```

25.9.2 Description

This command sets the mode of encapsulation that an existing RFC1483 transport uses.

25.9.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A
llc	Logical Link Control encapsulation method.	LLC
vcmux	VC Multiplexing encapsulation method.	

25.9.4 Example

```
prompt> rfc1483 set transport my1483 mode vcmux
```

25.9.5 See also

rfc1483 list transports on page 523

25.10 rfc1483 set transport pcr

25.10.1 Syntax

```
rfc1483 set transport {<name>|<number>} pcr <peak cell rate>
```

25.10.2 Description

This command sets the peak cell rate (pcr) for an existing RFC1483 transport.

25.10.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A
peak cell rate	Determines the maximum rate at which ATM cells are allowed to be transported into the ATM network. The PCR can be any value from 0 up to the maximum PortSpeed parameter set when the port was created (using the initbun file in FlashFS or the console command BUN *set port* on page 764).	0

25.10.4 Example

```
prompt> rfc1483 set transport my1483 pcr 50000
```

25.10.5 See also

rfc1483 set transport mcr on page 526

rfc1483 list transports on page 523

25.11 rfc1483 set transport port

25.11.1 Syntax

```
rfc1483 set transport {<name>|<number>} port <atm-port>
```

25.11.2 Description

This command sets the port that an existing RFC1483 transport uses to transport RFC1483 data over ATM. ATM ports are initialised in the initbun file in FlashFS, or using the *bun set port* console command.

25.11.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A
atm-port	The GlobespanVirata system port that is used to transport ATM data. The port/VPI/VCI combination must be unique for each transport.	N/A

25.11.4 Example

```
prompt> rfc1483 set transport my1483 port a1
```

25.11.5 See also

rfc1483 list transports on page 523

Console command BUN *set port* on page 764

25.12 rfc1483 set transport qosclass

25.12.1 Syntax

```
rfc1483 set transport {<name>|<number>} qosclass
{ubr|cbr|vbr|vbrrt|abr|qfc}
```

25.12.2 Description

This command sets the quality of service (QoS) class for the transport.

25.12.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A

Option	Description	Default value
ubr	Unspecified Bit Rate; non-constant and unpredictable data transport rate. PCR (Peak Cell Rate) is the average and maximum speed of transmission.	UBR
cbr	Constant Bit Rate; constant demand and predictable data transport rate. PCR is the average and maximum speed of transmission.	
vbr	Variable Bit Rate; non-constant but predictable data transport rate that uses Non-Real-Time (NRT). You can specify the PCR, SCR, BT and MBS for VBR traffic.	
vbrrt	Variable Bit Rate Real-Time; non-constant but predictable data transport rate that uses Real-Time (RT). You can specify the PCR, SCR, BT and MBS for VBRRT traffic.	
abr	Available Bit Rate; non-constant and unpredictable data transport rate that provides ATM-layer feedback and flow control.	
qfc	Quantum Flow Control; ATM flow control protocol that supports ABR.	

25.12.4 Example

```
prompt> rfc1483 set transport my1483 abr
```

25.12.5 See also

rfc1483 show transport on page 539

rfc1483 set transport bt on page 524

rfc1483 set transport mbs on page 525

rfc1483 set transport pcr on page 528

rfc1483 set transport scr on page 534

25.13 rfc1483 set transport rxvci

25.13.1 Syntax

```
rfc1483 set transport {<name>|<number>} rxvci <vci>
```

25.13.2 Description

This command sets the receive Virtual Circuit Identifier channel. If you later set the VCI using the *rfc1483 set transport vci* command, the RX VCI setting will be overridden.

The port/VCI/VPI combination must be unique for each transport.

25.13.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A
rxvci	Part of the ATM header. The RXVCI is a tag that identifies which channel a cell will be received over. The RXVCI can be any value between 1 and 65535.	VCI value set when the transport was created using the *rfc1483 add transport* command

25.13.4 Example

```
prompt> rfc1483 set transport my1483 rxvci 700
```

25.13.5 See also

rfc1483 list transports on page 523

rfc1483 set transport txvci on page 535

25.14 rfc1483 set transport rxvpi

```
rfc1483 set transport {<name>|<number>} rxvpi <vpi>
```

25.14.1 Description

This command sets the receive Virtual Path Identifier. If you later set the VPI using the *rfc1483 set transport vpi* command, the rxvpi setting will be overridden.

The port/VCI/VPI combination must be unique for each transport.

25.14.2 Options

The following table gives 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A
rxvpi	A field in the ATM header. The RXVPI is used to identify the virtual path that a circuit belongs to and receives information on. The RXVPI can be any value between 0 and 4095.	VPI value set when the transport was created using the *rfc1483 add transport* command

25.14.3 Example

```
prompt> rfc1483 set transport my1483 rxvpi 0
```

25.14.4 See also

rfc1483 list transports on page 523

rfc1483 set transport txvpi on page 536

25.15 rfc1483 set transport scr

25.15.1 Syntax

```
rfc1483 set transport {<name>|<number>} scr <sustainable cell rate>
```

25.15.2 Description

This command sets the Sustainable Cell Rate. This command is only valid if you set VBR or VBR RT as the QoS Class using the *rfc1483 set transport qosclass* command.

25.15.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A
sustainable cell rate	Sustainable Cell Rate; the average cell rate for a VBR or VBR RT connection. The SCR can be any positive value that is less than the both PortSpeed set (when the port was created) and the PCR set for the channel. (The port is initialised using the initbun file in FlashFS or the console command BUN *set port* on page 764.)	0

25.15.4 Example

```
prompt> rfc1483 set transport my1483 scr 25000
```

25.15.5 See also

rfc1483 set transport qosclass on page 530

rfc1483 list transports on page 523

25.16 rfc1483 set transport txvci

25.16.1 Syntax

```
rfc1483 set transport {<name>|<number>} txvci <vci>
```

25.16.2 Description

This command sets the transmit Virtual Circuit Identifier channel. If you later set the VCI using the *rfc1483 set transport vci* command, the txvci setting will be overridden.

The port/VCI/VPI combination must be unique for each transport.

25.16.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A
txvci	Part of the ATM header. The TX VCI is a tag that identifies which channel a cell will be transmitted over. The TX VCI can be any value between 1 and 65535.	VCI value set when the transport was created using the *rfc1483 add transport* command

25.16.4 Example

```
prompt> rfc1483 set transport my1483 txvci 800
```

25.16.5 See also

rfc1483 list transports on page 523

rfc1483 set transport rxvci on page 532

25.17 rfc1483 set transport txvpi

25.17.1 Syntax

```
rfc1483 set transport {<name>|<number>} txvpi <vpi>
```

25.17.2 Description

This command sets the transmit Virtual Path Identifier. If you later set the VPI using the *rfc1483 set transport vpi* command, the rxvpi setting will be overridden.

The port/VCI/VPI combination must be unique for each transport.

25.17.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A
txvpi	A field in the ATM header. The TX VPI is used to identify the virtual path that a circuit belongs to and transmits information on. The TX VPI can be any value between 0 and 4095.	VPI value set when the transport was created using the *rfc1483 add transport* command

25.17.4 Example

```
prompt> rfc1483 set transport my1483 txvpi 0
```

25.17.5 See also

rfc1483 list transports on page 523

rfc1483 set transport rxvpi on page 533

25.18 rfc1483 set transport vci

25.18.1 Syntax

```
rfc1483 set transport {<name>|<number>} vci <vci>
```

25.18.2 Description

This command sets the Virtual Circuit Identifier channel. It overrides existing VCI settings (including RX VCI and TX VCI).

The port/VCI/VPI combination must be unique for each transport.

25.18.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A
vci	Part of the ATM header. The VCI is a tag that identifies which channel a cell will travel over. The VCI can be any value between 1 and 65535.	VCI value set when the transport was created using the *add transport* command

25.18.4 Example

```
prompt> rfc1483 set transport my1483 vci 800
```

25.18.5 See also

rfc1483 list transports on page 523

rfc1483 set transport txvci on page 535

rfc1483 set transport rxvci on page 532

25.19 rfc1483 set transport vpi

25.19.1 Syntax

```
rfc1483 set transport {<name>|<number>} vpi <vpi>
```

25.19.2 Description

This command sets the Virtual Path Identifier. It overrides existing VPI settings (including RX VPI and TX VPI).

The port/VPI/VCI combination must be unique for each transport.

25.19.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A
vpi	A field in the ATM header. The VPI is used to identify the virtual path that a circuit belongs to. The VPI can be any value between 0 and 4095.	VPI value set when the transport was created using the *add transport* command

25.19.4 Example

```
prompt> rfc1483 set transport my1483 vpi 0
```

25.19.5 See also

rfc1483 list transports on page 523

rfc1483 set transport rxvpi on page 533

rfc1483 set transport txvpi on page 536

25.20 rfc1483 show transport

25.20.1 Syntax

```
rfc1483 show transport {<name>|<number>}
```

25.20.2 Description

This command displays the following information about an existing rfc1483 transport:

- Name
- Description
- Encapsulation method
- ATM port
- TX VPI - transmit Virtual Path Identifier
- RX VPI - receive Virtual Path Identifier
- TX VCI - transmit Virtual Circuit Identifier
- TX VCI - receive Virtual Circuit Identifier
- QOS class - Quality of Service class
- PCR - Peak Cell Rate
- SCR - Sustainable Cell Rate
- MCR - Minimum Cell Rate
- BT - Burst Tolerance
- MBS - Maximum Burst Size

25.20.3 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 RFC1483 transport. To display transport names, use the *rfc1483 list transports* command.	N/A
number	A number that identifies an existing RFC1483 transport. To display transport numbers, use the *rfc1483 list transports* command.	N/A

25.20.4 Example

```
prompt> rfc1483 show transport my1483

RFC1483 Transport: my1483

    Description: Default LAN port
  Encapsulation: LlcBridged

       ATM port: a1
         Tx VPI: 0
         Rx VPI: 0
         Tx VCI: 800
         Rx VCI: 800

      QOS class: UBR
           Peak cell rate: 0         Burst tolerance: 0
  Sustainable cell rate: 0         Max. burst size: 0
      Minimum cell rate: 0
```

25.20.5 See also

rfc1483 list transports on page 523

26.Security CLI commands

This chapter describes the Security CLI commands. The Security module is the parent module in the GlobespanVirata Security package.

For more information, see the ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS

26.1 Summary

26.1.1 About the VMI Security package

The VMI (GlobespanVirata Management Interface) **Security package** allows you to build security features into your GlobespanVirata system. The package contains three modules comprising of one parent module and two child modules:

1. The **Security module** is the *parent* module in the GlobespanVirata Security package. The Security commands allow you to:
   - enable/disable all modules in the GlobespanVirata Security package (including the child modules; NAT and Firewall)
   - add IP interfaces to the Security package to create security interfaces that are used to configure the NAT and Firewall child modules.
   - set triggers to allow applications to open secondary port sessions
2. The **NAT module** is a *child* module in the GlobespanVirata Security package. The NAT commands allow you to:
   - implement NAT based on the security interfaces added to the Security package by the *parent* Security module
   - enable NAT between an interface and an interface type
   - create global address pools
   - map global IP addresses to individual hosts on the inside network.
3. The stateful **Firewall module** is a *child* module in the GlobespanVirata Security package. The Firewall commands allow you to:
   - implement stateful Firewall features based on the security interfaces added in the *parent* Security module
   - set security levels and policies that determine whether traffic between interfaces is allowed or blocked
   - set portfilters to determine which ports and protocols are allowed or blocked
   - set IDSs (Intrusion Detection Settings).

For information on NAT commands, see *NAT CLI commands* on page 275.

For information on stateful Firewall commands, see *Firewall CLI commands* on page 173.

26.1.2 Security CLI commands

The table below lists the *security* commands provided by the CLI:

Command	Description/Console command
security enable|disable	*security* on page 545
security status	*security status* on page 546
security add interface	*security add interface* on page 547
security clear interfaces	*security clear interfaces* on page 548
security delete interface	*security delete interface* on page 549
security list interfaces	*security list interfaces* on page 550
security show interface	*security show interface* on page 551
security add trigger tcp|udp	*security add trigger tcp|udp* on page 552
security add trigger ftp|netmeeting	*security add trigger ftp|netmeeting* on page 554
security clear triggers	*security clear triggers* on page 556
security delete trigger	*security delete trigger* on page 557
security list triggers	*security list triggers* on page 558
security set trigger addressreplacement	*security set trigger addressreplacement* on page 559
security set trigger multihost	*security set trigger multihost* on page 561
security set trigger binaryaddressreplacement	*security set trigger binaryaddressreplacement* on page 562
security set trigger maxactinterval	*security set trigger maxactinterval* on page 563
security set trigger endport	*security set trigger endport* on page 564
security set trigger startport	*security set trigger startport* on page 565

Command	Description/Console command
security set trigger sessionchaining	*security set trigger sessionchaining* on page 566
security set trigger UDPsessionchaining	*security set trigger UDPsessionchaining* on page 567
security show trigger	*security show trigger* on page 569

26.2 security

26.2.1 Syntax

```
security {enable | disable}
```

26.2.2 Description

This command explicitly enables/disables all modules in the GlobespanVirata 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 GlobespanVirata system but want to save the security configuration between sessions, use the *system config save* command.

26.2.3 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
enabled	Enables all modules in the GlobespanVirata Security package (Security, NAT and Firewall modules).	disabled
disabled	Disables all modules in the GlobespanVirata Security package (Security, NAT and Firewall modules).	

26.2.4 Example

```
prompt> security enable
```

26.2.5 See also

firewall set securitylevel on page 182

system config save on page 598

26.3 security status

26.3.1 Syntax

```
security status
```

26.3.2 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)

26.3.3 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
```

26.3.4 See also

security on page 545

firewall set securitylevel on page 182

26.4 security add interface

26.4.1 Syntax

```
security add interface <name> {external|internal|dmz}
```

26.4.2 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.

26.4.3 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
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 de-militarized zone (DMZ)	N/A

26.4.4 Example

```
prompt> security add interface ip1 internal
```

26.4.5 See also

ip list interfaces on page 648

Firewall CLI commands on page 173

NAT CLI commands on page 275

26.5 security clear interfaces

26.5.1 Syntax

```
security clear interfaces
```

26.5.2 Description

This command removes all security interfaces that were added to the Security package using the *security add interface* command.

26.5.3 Example

```
prompt> security clear interfaces
```

26.5.4 See also

security delete interface on page 549

26.6 security delete interface

26.6.1 Syntax

```
security delete interface <name>
```

26.6.2 Description

This command removes a single security interface that was added to the Security package using the *security add interface* command.

26.6.3 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 security interface. To display interface names, use the *security list interfaces* command.	N/A

26.6.4 Example

```
prompt> security delete interface f1
```

26.6.5 See also

security clear interfaces on page 548

security list interfaces on page 550

26.7 security list interfaces

26.7.1 Syntax

```
security list interfaces
```

26.7.2 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)

26.7.3 Example

```
prompt> security list interfaces
Security Interfaces:
 ID   |   Name    | Type
-----|----------|----------
   1 | i1        | internal
   2 | i2        | external
   3 | i3        | dmz
---------------------------
```

26.7.4 See also

security show interface on page 551

26.8 security show interface

26.8.1 Syntax

```
security show interface <name>
```

26.8.2 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)

26.8.3 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
name	A name that identifies an existing security interface. To display all interface names, use the *security list interfaces* command.	N/A

26.8.4 Example

```
prompt> security show interface f2
Interface name: f2
Interface type: internal
```

26.8.5 See also

security list interfaces on page 550

26.9 security add trigger tcp|udp

26.9.1 Syntax

```
security add trigger <name> {tcp|udp} <startport>
<endport> <maxactinterval>
```

26.9.2 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* on page 218) 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.

26.9.3 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	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

Option	Description	Default value
endport	Sets the end of the trigger port range for the control session.	N/A
maxactinterval	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.	3000

26.9.4 Example

The following example creates an FTP (File Transfer Protocol) trigger:

```
prompt> security add trigger t1 tcp 21 21 3000
```

26.9.5 See also

security list triggers on page 558

security add trigger ftp|netmeeting on page 554

For more information on triggers, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

26.10 security add trigger ftp|netmeeting

26.10.1 Syntax

```
security add trigger <name> {ftp|netmeeting}
```

26.10.2 Description

This command allows you to use the example triggers provided by the CLI. It allows you to add a trigger to allow FTP or Netmeeting to transport data through the security package. Both applications open secondary port sessions. You do not have to set the port range or *maxactinterval* for FTP or Netmeeting triggers - the CLI automatically sets this for you.

26.10.3 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	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

Option	Description	Default value
ftp	Sets the trigger message format to ftp. The CLI automatically sets the following: Transport type = tcp Starting port = 21 Ending port = 21 Allow multiple hosts = false Max activity interval = 3000 Session chaining = false Session chaining on UDP = false Binary address replacement = false Address translation type = none	N/A
netmeeting	Sets the trigger message format to netmeeting. The CLI automatically sets the following: Transport type = tcp Starting port = 1720 Ending port = 1720 Allow multiple hosts = true Max activity interval = 30000 Session chaining = true Session chaining on UDP = false Binary address replacement = true Address translation type = tcp	N/A

26.10.4 Example

```
prompt> security add trigger t2 ftp
```

26.10.5 See also

security list triggers on page 558

security add trigger tcp|udp on page 552

For more information on triggers, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

26.11 security clear triggers

26.11.1 Syntax

```
security clear triggers
```

26.11.2 Description

This command deletes all triggers that were added to the Security module using the *security add trigger* commands.

26.11.3 Example

```
prompt> security clear triggers
```

26.11.4 See also

security delete trigger on page 557

26.12 security delete trigger

26.12.1 Syntax

```
security delete trigger <name>
```

26.12.2 Description

This command deletes a single trigger that was added to the Security module using the *security add trigger* commands.

26.12.3 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 trigger. To display trigger names, use the *security list trigger* command.	N/A

26.12.4 Example

```
prompt> security delete trigger t2
```

26.12.5 See also

security list triggers on page 558

security clear triggers on page 556

26.13 security list triggers

26.13.1 Syntax

```
security list triggers
```

26.13.2 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

26.13.3 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
-------------------------------------------
```

26.13.4 See also

security show trigger on page 569

26.14 security set trigger addressreplacement

26.14.1 Syntax

```
security set trigger <name> addressreplacement
{none|tcp|udp|both}
```

26.14.2 Description

The settings in this command are only effective if you enable address translation using the command *security set trigger binaryaddressreplacement* on page 562.

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.

26.14.3 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 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.	

26.14.4 Example

```
prompt> security set trigger t2 addressreplacement tcp
```

26.14.5 See also

security set trigger binaryaddressreplacement on page 562

For more information on address replacement, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

26.15 security set trigger multihost

26.15.1 Syntax

```
security set trigger <name> multihost {enable | disable}
```

26.15.2 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.

26.15.3 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 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.	

26.15.4 Example

```
prompt> security set trigger t1 multihost enable
```

26.15.5 See also

security list triggers on page 558

For more information on multihosts, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

26.16 security set trigger binaryaddressreplacement

26.16.1 Syntax

```
security set trigger <name> binaryaddressreplacement
{enable | disable}
```

26.16.2 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* on page 559.

26.16.3 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 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.	

26.16.4 Example

```
prompt> security set trigger t5 binaryaddressreplacement enable
```

26.16.5 See also

security set trigger addressreplacement on page 559

security list triggers on page 558

For more information on binary address replacement, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

26.17 security set trigger maxactinterval

26.17.1 Syntax

```
security set trigger <name> maxactinterval <interval>
```

26.17.2 Description

This command sets the maximum activity interval limit on existing session entries for an existing trigger.

26.17.3 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 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

26.17.4 Example

```
prompt> security set trigger t2 maxactinterval 5000
```

26.17.5 See also

security list triggers on page 558

26.18 security set trigger endport

26.18.1 Syntax

```
security set trigger <name> endport <portnumber>
```

26.18.2 Description

This command sets the end of the port number range for an existing trigger.

26.18.3 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 trigger. To display trigger names, use the *security list triggers* command.	N/A
portnumber	Sets the end of the trigger port range.	N/A

26.18.4 Example

```
prompt> security set trigger t3 endport 21
```

26.18.5 See also

security set trigger startport on page 565

26.19 security set trigger startport

26.19.1 Syntax

```
security policy <name> set trigger startport <portnumber>
```

26.19.2 Description

This command sets the start of the port number range for an existing trigger.

26.19.3 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 trigger. To display trigger names, use the *security list triggers* command.	N/A
port	Sets the start of the trigger port range.	N/A

26.19.4 Example

```
prompt> security set trigger t3 startport 21
```

26.19.5 See also

security set trigger endport on page 564

26.20 security set trigger sessionchaining

26.20.1 Syntax

```
security set trigger <name> sessionchaining {enable | disable}
```

26.20.2 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.

26.20.3 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 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.	

26.20.4 Example

```
prompt> security set trigger t4 sessionnchaining enable
```

26.20.5 See also

security set trigger UDPsessionchaining on page 567

For more information on session chaining, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

26.21 security set trigger UDPsessionchaining

26.21.1 Syntax

```
security set trigger <name> UDPsessionchaining {enable | disable}
```

26.21.2 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.

26.21.3 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 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.	

26.21.4 Example

```
prompt> security set trigger t3 UDPsessionchaining enable
```

26.21.5 See also

security set trigger sessionchaining on page 566

For more information on UDP session chaining, see the *ISOS Security (NAT and Firewall) Functional Specification: DO-008557-PS*.

26.22 security show trigger

26.22.1 Syntax

```
security show trigger <name>
```

26.22.2 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)

26.22.3 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 trigger. To display trigger names, use the *security list triggers* command.	N/A

26.22.4 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
```

26.22.5 See also

security list triggers on page 558

27.SNTP CLI commands

This chapter describes the SNTP (Simple Network Time Protocol) CLI commands.

27.1 Summary

27.1.1 SNTP CLI commands

The table below lists the SNTP commands provided by the CLI:

Command	Reference
sntpclient set timezone	*sntpclient set timezone* on page 573
sntpclient set mode	*sntpclient set mode* on page 577
sntpclient set server	*sntpclient set server* on page 579
sntpclient set poll-interval	*sntpclient set poll-interval* on page 580
sntpclient sync	*sntpclient sync* on page 581
sntpclient set timeout	*sntpclient set timeout* on page 582
sntpclient set retries	*sntpclient set retries* on page 583
sntpclient show association	*sntpclient show association* on page 584
sntpclient show status	*sntp show status* on page 585
sntpclient set clock	*sntpclient set clock* on page 586

27.2 sntpclient set timezone

27.2.1 Syntax

```
sntpclient set timezone <abbreviation>
```

27.2.2 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).

27.2.3 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:

Abbreviation	± UTC	World Area of Time Zone
IDLW	-1200	International Date Line West
NT	-1100	Nome
HST	-1000	Hawaii Standard
CAT	-0900	Central Alaska
AHST	-0900	Alaska-Hawaii Standard
YST	-0900	Yukon Standard
HDT	-0900	Hawaii Daylight
YDT	-0800	Yukon Daylight
PST	-0800	US Pacific Standard
PDT	-0700	US Pacific Daylight
MST	-0700	US Mountain Standard
MDT	-0600	US Mountain Daylight
CST	-0600	US Central Standard

Abbreviation	± UTC	World Area of Time Zone
CDT	-0500	US Central Daylight
EST	-0500	US Eastern Standard
EDT	-0400	US Eastern Daylight
AST	-0400	Atlantic Standard
NFST	-0330	Newfoundland Standard
NFT	-0330	Newfoundland
BRST	-0300	Brazil Standard
ADT	-0300	Atlantic Daylight
NDT	-0230	Newfoundland Daylight
AT	-0200	Azores
WAT	-0100	West Africa
GMT	+0000	Greenwich Mean
UTC	+0000	Universal (Coordinated)
WET	+0000	Western European
CET	+0100	Central European
FWT	+0100	French Winter
MET	+0100	Middle European
MEWT	+0100	Middle European Winter
SWT	+0100	Swedish Winter
BST	+0100	British Summer
EET	+0200	Eastern Europe, USSR Zone 1
FST	+0200	French Summer
MEST	+0200	Middle European Summer
SST	+0200	Swedish Summer
IST	+0200	Israeli Standard
IDT	+0300	Israeli Daylight

Abbreviation	± UTC	World Area of Time Zone
BT	+0300	Baghdad, USSR Zone 2
IT	+0330	Iran
ZP4	+0400	USSR Zone 3
ZP5	+0500	USSR Zone 4
INST	+0530	Indian Standard
ZP6	+0600	USSR Zone 5
NST	+0630	North Sumatra
WAST	+0700	West Australian Standard
SSMT	+0700	South Sumatra, USSR Zone 6
JT	+0730	Java
CCT	+0800	China Coast, USSR Zone 7
WADT	+0800	West Australian Daylight
ROK	+0900	Korean Standard
KST	+0900	Korean Standard
JST	+0900	Japan Standard, USSR Zone 8
CAST	+0930	Central Australian Standard
KDT	+1000	Korean Daylight
EAST	+1000	Eastern Australian Standard
GST	+1000	Guam Standard, USSR Zone 9
CADT	+1030	Central Australian Daylight
EADT	+1100	Eastern Australian Daylight
IDLE	+1200	International Date Line East
NZST	+1200	New Zealand Standard
NZT	+1200	New Zealand
NZDT	+1300	New Zealand Daylight

27.2.4 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
```

27.3 sntpclient set mode

27.3.1 Syntax

```
sntpclient set mode {unicast|broadcast|anycast} {enable|disable}
```

27.3.2 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.

27.3.3 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
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	Enables the selected time synchronous access mode.	N/A

27.3.4 Example

```
prompt> sntpclient set mode anycast enable
```

27.3.5 See also

sntpclient set server on page 579

sntp show status on page 585

sntpclient show association on page 584

27.4 sntpclient set server

27.4.1 Syntax

```
sntpclient set server {ipaddress <IP address> | hostname <hostname>}
```

27.4.2 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.

27.4.3 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
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

27.4.4 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
```

27.5 sntpclient set poll-interval

27.5.1 Syntax

```
sntpclient set poll-interval <0-30>
```

27.5.2 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.

27.5.3 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
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 (disabled)

27.5.4 Example

```
prompt> sntpclient set poll-interval 10
```

27.6 sntpclient sync

27.6.1 Syntax

```
sntpclient sync
```

27.6.2 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.

27.6.3 Example

```
prompt> sntpclient sync
```

27.6.4 See also

sntpclient set server on page 579

27.7 sntpclient set timeout

27.7.1 Syntax

```
sntpclient set timeout <0-30>
```

27.7.2 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.

27.7.3 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
0-30	Sets the received packet response timeout value (in seconds). This can be any value between 0 and 30.	5 seconds

27.7.4 Example

```
prompt> sntpclient set timeout 10
```

27.7.5 See also

sntpclient set retries on page 583

27.8 sntpclient set retries

27.8.1 Syntax

```
sntpclient set retries <0-10>
```

27.8.2 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.

27.8.3 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
0-10	Sets the number of packet retry attempts made when no response is received from a timeserver.	2

27.8.4 Example

```
prompt> sntpclient set retries 4
```

27.8.5 See also

sntpclient set timeout on page 582

27.9 sntpclient show association

27.9.1 Syntax

```
sntpclient show association
```

27.9.2 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.

27.9.3 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.
```

27.9.4 See also

sntpclient set server on page 579

27.10 sntp show status

27.10.1 Syntax

```
sntpclient show status
```

27.10.2 Description

This command displays the SNTP client status information.

27.10.3 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
```

27.10.4 See also

sntpclient show association on page 584

27.11 sntpclient set clock

27.11.1 Syntax

```
sntpclient set clock <yyyy:mm:dd:hh:mm:ss>
```

27.11.2 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.

27.11.3 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
```

28.System CLI commands

This chapter describes the System CLI commands.

28.1 Summary

28.1.1 System CLI commands

The table below lists the *system* commands provided by the CLI:

Command	Description/Console command
system add user	*system add user* on page 591
system add login	*system add login* on page 593
system config backup	*system config backup* on page 595
system config restore	*system config restore* on page 596
system config save	*system config save* on page 598
system delete login	*system delete login* on page 599
system delete user	*system delete user* on page 600
system info	*system info* on page 601
system list errors	*system list errors* on page 602
system list openfiles	*system list openfiles* on page 603
system list users	*system list users* on page 604
system list logins	*system list logins* on page 605
system log	*system log* on page 606
system log enable|disable	*system log enable|disable* on page 607
system log list	*system log list* on page 609
system restart	*system restart* on page 611
system set login access	*system set login access* on page 612
system set login mayconfigure	*system set login mayconfigure* on page 613
system set login maydialin	*system set login maydialin* on page 614
system set user access	*system set user access* on page 615
system set user mayconfigure	*system set user mayconfigure* on page 616
system set user maydialin	*system set user maydialin* on page 617

28.1.2 System Console commands

The table below lists the *top-level* and *configuration* **console** commands and, if available, their equivalent CLI command:

Command	CLI Equivalent
restart	Replaced by CLI command *system restart* on page 611
version	Usable command, see *version* on page 50
event	Usable command, see *event …* on page 707
. (history mechanism)	Usable command, see *. (history mechanism)* on page 715
list	Usable command, see *list* on page 719
echo	Usable command, see *echo …* on page 721
buildid	Usable command, see *buildid* on page 727
help	Usable command, see *help* on page 775
update	Usable command
@ commands	Blacklisted command, see *@ commands* on page 716
<process>,<process>,<command>	Blacklisted command, see *<process>, <process> <command>* on page 712
tell	Blacklisted command, see *tell <process> …* on page 722
exit, exit!	Blacklisted command, see *exit, exit!* on page 723
debug	Blacklisted command, see *debug* on page 724
crlf, nocrlf	Blacklisted command, see *crlf, nocrlf* on page 725
bind, unbind	Blacklisted command, see *bind <process>, unbind* on page 726
config list	Blacklisted command, see *list* on page 719
config print	Blacklisted command

Command	CLI Equivalent
config reset	Replaced by CLI command *system config restore* on page 596
config save	Replaced by CLI command *system config save* on page 598
config resource	Blacklisted command
config version	Blacklisted command, see *version* on page 50

28.2 system add user

28.2.1 Syntax

```
system add user <name> ["comment"]
```

28.2.2 Description

This command adds a user (typically a PPP user) to the system. Only a Super user can use this command.

28.2.3 Default setting

The default setting in the table below are applied to new accounts added using the *system add user* command. (A different set of defaults are applied to a new account added using the *system add login* command.)

Option	Default Setting
dialin to the system	enabled
login to the system	disabled
configuration permissions	disabled
access permissions	default user

28.2.4 Options

The following table gives 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* on page 604 and *system list logins* on page 605.	No comment added

28.2.5 Example

```
prompt> system add user <name> ["comment"]
```

28.2.6 See also

system set user access on page 615

system set user maydialin on page 617

system set user mayconfigure on page 616

system list users on page 604

system delete user on page 600

28.3 system add login

28.3.1 Syntax

```
system add login <name> ["comment"]
```

28.3.2 Description

This command adds a user to the system. Only a Super user can use this command.

28.3.3 Default setting

The default setting in the table below are applied to new accounts added using the *system add login* command. (A different set of defaults are applied to a new account added using the *system add user* command.)

Option	Default Setting
dialin to the system	disabled
login to the system	enabled
configuration permissions	enabled
access permissions	default user

28.3.4 Options

The following table gives 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
comment	An optional comment about the user that is displayed when you type the command *system list users* on page 604 and *system list logins* on page 605.	Blank (No comment added)

28.3.5 Example

```
prompt> system add login ckearns "temporary contractor"
```

28.3.6 See also

system delete login on page 599

system list logins on page 605

28.4 system config backup

28.4.1 Syntax

```
system config backup [filename]
```

28.4.2 Description

This command saves the system configuration to a file. To specify the file that you want to save configuration information in, type a *filename* value. If you do not specify a filename, the configuration is saved in the *im.conf.backup* file by default.

To prevent a user from overwriting the system with their own configuration, only a Super user can use this command.

28.4.3 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
filename	The name of an existing file that you want to save your configuration in.	If a filename is not specified, the configuration is saved in *im.conf.backup*.

28.4.4 Example

```
prompt> system config backup

Saving to backup configuration //isfs/im.conf backup
```

28.4.5 See also

system config restore on page 596

system config save on page 598

28.5 system config restore

28.5.1 Syntax

```
system config restore {backup [filename]|factory}
```

28.5.2 Description

This command allows Super users, Engineers and Default users to restore their backup configuration. By default, backups are restored from the *im.conf.backup* file. If your configuration is saved in a different file, you can restore your backup configuration by typing a *filename* value. Note that **only** Super users can backup their system configuration to a specified filename. See *system config backup* on page 595.

The 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.

This command also allows Super users to restore the factory defaults from *im.conf.factory*.

28.5.3 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
backup	Restores the backup configuration from the *im.conf.backup* file.	N/A
factory	Restores the factory default configuration from the *im.conf.factory* file. Only Super users can use this command.	N/A

28.5.4 Example

```
prompt> system config restore backup
Restoring backup configuration //isfs/im.conf backup
```

28.5.5 See also

system config backup on page 595

system config save on page 598

28.6 system config save

28.6.1 Syntax

```
system config save
```

28.6.2 Description

This command saves the system configuration in the *im.conf* file in FlashFS. This allows all users to create their own backup files. Default, Engineer and Super users can use this command.

28.6.3 Example

```
prompt> system config save
Wait for 'configurataion saved' message...
-->
Saving configuration...
Configuration saved.
```

28.6.4 See also

system config backup on page 595

system config restore on page 596

Super users can list all configuration files using the console command *fm ls*. For more information, see *The ATMOS File Manager Process Functional Specification: DO-008609-PS* and the *Using File Manager* chapter in the Software User's Guide for the GlobespanVirata system that you are using.

28.7 system delete login

28.7.1 Syntax

```
system delete login <name>
```

28.7.2 Description

This command deletes a user that has been added to the system using the *system add login* command. Only a Super user can use this command.

28.7.3 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	The name of an existing user.	N/A

28.7.4 Example

```
prompt> system delete login ckearns
```

28.7.5 See also

system add login on page 593

28.8 system delete user

28.8.1 Syntax

```
system delete user <name>
```

28.8.2 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 a Super user can use this command.

28.8.3 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
name	The name of an existing user.	N/A

28.8.4 Example

```
prompt> system delete user ckearns
```

28.8.5 See also

system add user on page 591

system add login on page 593

28.9 system info

28.9.1 Syntax

```
system info
```

28.9.2 Description

This command displays the vendor ID, URL, base MAC address and hardware and software version details of the GlobespanVirata system that you are using.

28.9.3 Example

```
prompt> system info
Global System Configuration:

           vendor:  GlobespanVirata
              URL:  http://www.globespanvirata.com

      MAC address:  ##:##:##:##:##:##

 Hardware version:  BD3000 BSP v2.0 (ISOS 8.0)
 Software version:  8.0.0.25
```

28.9.4 See also

To obtain more detailed information about the system, see the *isos-ver* command contained in the latest tools release.

28.10 system list errors

28.10.1 Syntax

```
system list errors
```

28.10.2 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

28.10.3 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)
--------------------------------------------------------------------------------
```

28.10.4 See also

system list users on page 604

system list logins on page 605

28.11 system list openfiles

28.11.1 Syntax

```
system list openfiles <name>
```

28.11.2 Description

This command allows you to display low-level debug information about specific open file handles.

28.11.3 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	The name of a file which has open file handles associated with it.	N/A

28.11.4 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
```

28.11.5 See also

system log enable|disable on page 607

28.12 system list users

28.12.1 Syntax

```
system list users
```

28.12.2 Description

This command displays a list of users and logins added to the system using the *system add user* and *system add login* commands. The same information is displayed by the *system list logins* command.

The list contains the following information:

- user ID number
- user name
- configuration permissions (enabled or disabled)
- dialin permissions (enabled or disabled)
- access level (default, engineer or super user)
- comment (any comments that were included when the user was added to the system)

28.12.3 Example

```
prompt> system list users
Users:
                      May        May       Access
 ID  |   Name     |  Conf.   |  Dialin  |   Level    |  Comment
-----|------------|----------|----------|------------|---------------------
   1 | admin      | ENABLED  | disabled | superuser  | Default admin user
-------------------------------------------------------------------------
```

28.12.4 See also

system list errors on page 602

system list logins on page 605

28.13 system list logins

28.13.1 Syntax

```
system list logins
```

28.13.2 Description

This command displays a list of logins and users added to the system using the *system add login* and *system add user* commands. The same information is displayed by the *system list users* command.

The list contains the following information:

- user ID number
- user name
- configuration permissions (enabled or disabled)
- dial in permissions (enabled or disabled)
- access level (default, engineer or super user)
- comment (any comments that were included when the user was added to the system)

28.13.3 Example

```
prompt> system list logins
Users:
                      May         May        Access
 ID  |   Name      |  Conf.    |  Dialin  |   Level     |  Comment
-----|-------------|-----------|----------|-------------|--------------------
   1 | admin       | ENABLED   | disabled | superuser   | Default admin user
-------------------------------------------------------------------------------
```

28.13.4 See also

system list errors on page 602

system list users on page 604

28.14 system log

28.14.1 Syntax

```
system log {nothing|warnings|info|trace|entryexit|all}
```

28.14.2 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.

28.14.3 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
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

28.14.4 Example

```
prompt> system log all
```

28.15 system log enable|disable

28.15.1 Syntax

```
system log {enable|disable} rip {errors|rx|tx}
system log {enable|disable} ip {icmp|rawip|udp|tcp|arp|socket}
system log {enable|disable} ipoa {debug|trace}
```

28.15.2 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*, *IP* and *IPoA*.

The command only supports modules that are present in the current image that you are using. For example, if you do not have IPoA included in your image build, the IPoA options are not available.

Each individual module has its own specific module category (see *Examples* on page 608). 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.

The IPoA *trace* and *debug* categories *are* related to each other as follows:

- *trace* provides useful general output which shows what is happening in the IPoA module. Enabling *trace* turns trace on.
- *debug* provides detailed output that is useful for debugging detailed problems. Enabling *debug* turns on debug and trace output.

To display a list of modules and categories and their enable/disable status, see *system log list* on page 609.

28.15.3 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
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.	

28.15.4 Examples

Example One - RIP

```
prompt> system log enable rip rx
enabled logging for rip rx
```

Example Two - IPoA

```
prompt> system log enable ipoa debug
enabled logging for ipoa debug
```

28.15.5 See also

system log list on page 609

system log on page 606

28.16 system log list

28.16.1 Syntax

```
system log list [<module>]
```

28.16.2 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. Both commands displays the current status of the tracing options set using the command *system log enable|disable* on page 607.

The command only displays modules and categories that are present in the current image that you are using. For example, if you do not have IPoA included in your image build, the IPoA options are not listed in the output for this command.

28.16.3 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
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

28.16.4 Examples

Example One - *system log list*

```
prompt> system log list
 rip   errors (ENABLED)
 rip       rx (disabled)
 rip       tx (disabled)
  ip     icmp (disabled)
  ip    rawip (ENABLED)
```

```
  ip      udp (disabled)
  ip      tcp (disabled)
  ip      arp (disabled)
  ip   socket (disabled)
ipoa    debug (ENABLED)
ipoa    trace (ENABLED)
```

Example Two - *system log list <module>*

```
prompt> system log list ip
  ip     icmp (disabled)
  ip    rawip (ENABLED)
  ip      udp (disabled)
  ip      tcp (disabled)
  ip      arp (disabled)
  ip   socket (disabled)
```

28.16.5 See also

system log on page 606

system log enable|disable on page 607

28.17 system restart

28.17.1 Syntax

```
system restart
```

28.17.2 Description

This command restarts your GlobespanVirata system. It has the same effect as pressing the reset button on your Virata ISOS system.

28.17.3 Example

```
prompt> system restart
```

28.17.4 See also

system config restore on page 596

28.18 system set login access

28.18.1 Syntax

```
system set login <name> access {default|engineer|superuser}
```

28.18.2 Description

This command sets the access permissions of a user who has been added to the system using the *system add login* command. Only a Super user can use this command.

28.18.3 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	The name of an existing user.	N/A
default/ engineer/ superuser	Access permissions for a user.	default

28.18.4 Example

```
prompt> system set login ckearns access engineer
```

28.18.5 See also

system set login mayconfigure on page 613

system set login maydialin on page 614

For more information on the types of user access permissions, see *Access permissions to the CLI* on page 9.

28.19 system set login mayconfigure

28.19.1 Syntax

```
system set login <name> mayconfigure {enabled|disabled}
```

28.19.2 Description

This command sets configuration permissions for a user who has been added to the system using the *add system login* or the *add system user* command. Only a Super user can use this command.

Note - If you 'disable' your own login configuration permissions and then logout of the system, you **will not** be able to login again. However, there is a security measure that allows you to access the system without a password **if** you were the last Super user to logout of the system.

28.19.3 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	The name of an existing user.	N/A
enabled/ disabled	Determines whether or not a user can configure the system.	enabled

28.19.4 Example

```
prompt> system set login ckearns mayconfigure disabled
```

28.19.5 See also

system set login access on page 612

system set login maydialin on page 614

28.20 system set login maydialin

28.20.1 Syntax

```
system set login <name> maydialin {enabled|disabled}
```

28.20.2 Description

This command sets dial in permissions for a user who has been added to the system using the *system add login* command. Only a Super user can use this command.

28.20.3 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	The name of an existing user.	N/A
enabled/ disabled	Determines whether or not a user can dialin to the system.	disabled

28.20.4 Example

```
prompt> system set login ckearns maydialin enabled
```

28.20.5 See also

system set login access on page 612

system set login mayconfigure on page 613

28.21 system set user access

28.21.1 Syntax

```
system set user <name> access {default|engineer|superuser}
```

28.21.2 Description

This command sets the access permissions of a user who has been added to the system using the *system add user* command. Only a Super user can use this command.

28.21.3 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	The name of an existing user.	N/A
default/ engineer/ superuser	Allows you to set the access permissions for a user.	default

28.21.4 Example

```
prompt> system set user ckearns access default
```

28.21.5 See also

system set user mayconfigure on page 616

system set user maydialin on page 617

For more information on the different types of user access permissions, see *Access permissions to the CLI* on page 9.

28.22 system set user mayconfigure

28.22.1 Syntax

```
system set user <name> mayconfigure {enabled|disabled}
```

28.22.2 Description

This command sets configuration permissions for a user who has been added to the system using the *add system user* command. Only a Super user can use this command.

28.22.3 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	The name of an existing user.	N/A
enabled/ disabled	Determines whether or not a user can configure the system.	disabled

28.22.4 Example

```
prompt> system set user ckearns mayconfigure enabled
```

28.22.5 See also

system set user access on page 615

system set user maydialin on page 617

28.23 system set user maydialin

28.23.1 Syntax

```
system set user <name> maydialin {enabled|disabled}
```

28.23.2 Description

This command sets dial in permissions for a user who has been added to the system using the *system add user* command. Only a Super user can use this command.

28.23.3 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	The name of an existing user.	N/A
enabled/ disabled	Determines whether or not a user can dialin to the system.	enabled

28.23.4 Example

```
prompt> system set user ckearns maydialin enabled
```

28.23.5 See also

system set user access on page 615

system set user mayconfigure on page 616

29.TCP/IP CLI commands

This chapter describes the TCP/IP CLI commands.

For information on the IP stack, see the IP Stack Programmer's Guide: DO-010017-TC.

29.1 Summary

29.1.1 IP Tracing commands

You can carry out tracing in the IP stack using the following *system* commands:

- *system log enable|disable* on page 607; enables/disables the tracing support output for a specific module and category.
- *system log list* on page 609; displays the tracing options for the modules available in your current image.

29.1.2 IP CLI commands

The table below lists the *IP* commands provided by the CLI:

Command	Reference
ip add interface	*ip add interface* on page 623
ip add route	*ip add route* on page 626
ip add defaultroute gateway	*ip add defaultroute gateway* on page 629
ip add defaultroute interface	*ip add defaultroute interface* on page 630
ip attachbridge	*ip attachbridge* on page 631
ip attach	*ip attach* on page 632
ip clear interfaces	*ip clear interfaces* on page 634
ip clear riproutes	*ip clear riproutes* on page 635
ip clear routes	*ip clear routes* on page 636
ip delete interface	*ip delete interface* on page 637
ip delete route	*ip delete route* on page 638
ip detach interface	*ip detach interface* on page 639
ip interface add secondaryipaddress	*ip interface add secondaryipaddress* on page 640
ip interface clear secondaryipaddress	*ip interface clear secondaryipaddresses* on page 642
ip interface delete secondaryipaddress	*ip interface delete secondaryipaddress* on page 643

Command	Reference
ip interface list secondaryipaddresses	*ip interface list secondaryipaddresses* on page 644
ip list arpentries	*ip list arpentries* on page 646
ip list connections	*ip list connections* on page 647
ip list interfaces	*ip list interfaces* on page 648
ip list riproutes	*ip list riproutes* on page 649
ip list routes	*ip list routes* on page 650
ip ping	*ip ping* on page 651
ip set interface ipaddress	*ip set interface ipaddress* on page 652
ip set netmask	*ip set interface netmask* on page 654
ip set interface mtu	*ip set interface mtu* on page 655
ip set interface dhcp	*ip set interface dhcp* on page 656
ip set interface rip accept	*ip set interface rip accept* on page 657
ip set interface rip multicast	*ip set interface rip multicast* on page 659
ip set interface tcpmssclamp	*ip set interface tcpmssclamp* on page 663
ip set interface rip send	*ip set interface rip send* on page 661
ip set rip hostroutes	*ip set rip hostroutes* on page 664
ip set rip poison	*ip set rip poison* on page 665
ip set route destination	*ip set route destination* on page 666
ip set route gateway	*ip set route gateway* on page 667
ip set route cost	*ip set route cost* on page 669
ip set route interface	*ip set route interface* on page 670
ip show	*ip show* on page 671
ip show interface	*ip show interface* on page 672
ip show route	*ip show route* on page 674

29.1.3 IP Console commands

The table below lists the *ip* **console** commands and, if available, their CLI equivalent command:

Command	CLI status
ip config	Replaced by CLI command *ip show* on page 671 and *ip list interfaces* on page 648
ip device add	Replaced by CLI command *ip add interface* on page 623
ip device delete	Replaced by CLI command *ip delete interface* on page 637
ip device flush	Blacklisted command, see *device* on page 833
ip device list	Replaced by CLI command *ip list interfaces* on page 648
ip enable	Replaced by CLI command *ip attach* on page 632.
ip help	Usable command, see *help* on page 838
ip portname	Blacklisted command, see *portname* on page 840
ip snmp access list	Usable command, see *snmp* on page 842
ip snmp access delete	Blacklisted command, see *snmp* on page 842
ip snmp access flush	Blacklisted command, see *snmp* on page 842
ip snmp access read|write	Blacklisted command, see *snmp* on page 842
ip snmp config	Usable command, see *snmp* on page 842
ip snmp config save	Blacklisted command, see *snmp* on page 842
ip snmp trap add	Blacklisted command, see *snmp* on page 842
ip snmp trap delete	Blacklisted command, see *snmp* on page 842
ip snmp trap list	Usable command, see *snmp* on page 842
ip version	Replaced by CLI command *ip show* on page 671.

29.2 ip add interface

29.2.1 Syntax

```
ip add interface <name> [<ipaddress> [<netmask>]]
```

29.2.2 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. See *PPPoA CLI commands* on page 311, *PPPoE CLI commands* on page 383 or *PPPoH CLI commands* on page 451.

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. For more details, see *Options* on page 624.

29.2.3 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	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.

29.2.4 Example

```
prompt> ip add interface ip1 192.168.103.3 255.255.255.0
```

29.2.5 See also

ip attach on page 632

ip show interface on page 672

ip set interface ipaddress on page 652

ip set interface dhcp on page 656

For information on setting DHCP client configuration options, see *DHCP Client CLI commands* on page 61.

29.3 ip add route

29.3.1 Syntax

```
ip add route <name> <dest_ip> <netmask> {[gateway
<gateway_ip>]|[interface <interface>]}
```

29.3.2 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.

Note that 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.

29.3.3 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	An arbitrary name that identifies the 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

29.3.4 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
```

29.3.5 See also

ip list interfaces on page 648

ip add defaultroute gateway on page 629

ip add defaultroute interface on page 630

29.4 ip add defaultroute gateway

29.4.1 Syntax

```
ip add defaultroute gateway <gateway_ip>
```

29.4.2 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
```

Note that 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.

29.4.3 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
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

29.4.4 Example

```
prompt> ip add defaultroute gateway 192.168.103.3
```

29.4.5 See also

ip add route on page 626

ip add defaultroute interface on page 630

29.5 ip add defaultroute interface

29.5.1 Syntax

```
ip add defaultroute interface <interface>
```

29.5.2 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
```

Note that 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.

29.5.3 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
interface	The name of the existing interface that this route will use. To display interface names, use the *ip list interfaces* command.	N/A

29.5.4 Example

```
prompt> ip add defaultroute interface ip3
```

29.5.5 See also

ip add route on page 626

ip add defaultroute gateway on page 629

29.6 ip attachbridge

29.6.1 Syntax

```
ip attachbridge {<name>|<number>}
```

29.6.2 Description

This command attaches the bridge to the router via an existing IP interface.

29.6.3 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

29.6.4 Example

```
prompt> ip attachbridge 2
```

29.6.5 See also

ip add interface on page 623

ip list interfaces on page 648

For more information on transports and interfaces, see the User's Guide for your GlobespanVirata system.

29.7 ip attach

29.7.1 Syntax

```
ip attach {<name>|<number>} <transport>
```

29.7.2 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.

29.7.3 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
transport	A name that identifies an existing transport.	N/A

29.7.4 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`**

29.7.5 See also

ip add interface on page 623

ip list interfaces on page 648

For information on a particular transport type, see the relevant transport chapter:

PPPoA CLI commands on page 311

PPPoE CLI commands on page 383

Frame Relay CLI commands on page 223

IPoA CLI commands on page 245

Ethernet CLI commands on page 163

RFC1483 CLI commands on page 517

29.8 ip clear interfaces

29.8.1 Syntax

```
ip clear interfaces
```

29.8.2 Description

This command clears all IP interfaces that were created using the *ip add interface* command.

29.8.3 Example

```
prompt> ip clear interfaces
```

29.8.4 See also

ip delete interface on page 637

29.9 ip clear riproutes

29.9.1 Syntax

```
ip clear riproutes
```

29.9.2 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.

29.9.3 Example

```
prompt> ip clear riproutes
```

29.9.4 See also

ip clear routes on page 636

ip set rip hostroutes on page 664

ip set interface rip accept on page 657

ip set interface rip send on page 661

29.10 ip clear routes

29.10.1 Syntax

```
ip clear routes
```

29.10.2 Description

This command clears all static routes that were created using the *ip add route* command.

29.10.3 Example

```
prompt> ip clear routes
```

29.10.4 See also

ip delete route on page 638

29.11 ip delete interface

29.11.1 Syntax

```
ip delete interface {<name>|<number>}
```

29.11.2 Description

This command deletes a single IP interface that was created using the *ip add interface* command.

29.11.3 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

29.11.4 Example

```
prompt> ip delete interface ip1
```

29.11.5 See also

ip clear interfaces on page 634

ip list interfaces on page 648

29.12 ip delete route

29.12.1 Syntax

```
ip delete route {<name>|<number>}
```

29.12.2 Description

This command deletes a single route that was created using the *ip add route* command.

29.12.3 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 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

29.12.4 Example

```
prompt> ip delete route route1
```

29.12.5 See also

ip list routes on page 650

29.13 ip detach interface

29.13.1 Syntax

```
ip detach {<name>|<number>}
```

29.13.2 Description

This command detaches a transport from an IP interface which was previously attached using the *ip attach interface* command.

29.13.3 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

29.13.4 Example

```
prompt> ip detach ip1
```

29.13.5 See also

ip list interfaces on page 648

29.14 ip interface add secondaryipaddress

29.14.1 Syntax

```
ip interface {<name>|<number>} add secondaryipaddress
<ipaddress> [<netmask>]
```

29.14.2 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.

29.14.3 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
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

29.14.4 Example

```
prompt> ip interface ip1 add secondaryipaddress 192.168.102.3 255.255.255.0
```

29.14.5 See also

ip list interfaces on page 648

ip interface list secondaryipaddresses on page 644

29.15 ip interface clear secondaryipaddresses

29.15.1 Syntax

```
ip interface {<name>|<number>} clear secondaryipaddresses
```

29.15.2 Description

This command deletes all additional IP addresses that have been added to an existing IP interface using the *ip interface add secondaryipaddress* command.

29.15.3 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

29.15.4 Example

```
prompt> ip interface ip1 clear secondaryipaddresses
```

29.15.5 See also

ip list interfaces on page 648

ip interface add secondaryipaddress on page 640

ip interface delete secondaryipaddress on page 643

ip interface list secondaryipaddresses on page 644

29.16 ip interface delete secondaryipaddress

29.16.1 Syntax

```
ip interface {<name>|<number>} delete secondaryipaddress
<secondaryipaddress number>
```

29.16.2 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.

29.16.3 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
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

29.16.4 Example

```
prompt> ip interface ip1 delete secondaryipaddress 1
```

29.16.5 See also

ip list interfaces on page 648

ip interface list secondaryipaddresses on page 644

29.17 ip interface list secondaryipaddresses

29.17.1 Syntax

```
ip interface {<name>|<number>} list secondaryipaddresses
```

29.17.2 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.

29.17.3 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

29.17.4 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
---------------------------------------
```

29.17.5 See also

ip list interfaces on page 648

ip interface list secondaryipaddresses on page 644

29.18 ip list arpentries

29.18.1 Syntax

```
ip list arpentries
```

29.18.2 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.

29.18.3 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
-------------------------------------------------------------
```

29.19 ip list connections

29.19.1 Syntax

```
ip list connections
```

29.19.2 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.

29.19.3 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                  | *:*                    |
-------------------------------------------------------------------------
```

29.20 ip list interfaces

29.20.1 Syntax

```
ip list interfaces
```

29.20.2 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

29.20.3 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 | 0.0.0.0
-------------------------------------------------------------------
```

29.20.4 See also

ip show interface on page 672

ip set interface dhcp on page 656

29.21 ip list riproutes

29.21.1 Syntax

```
ip list riproutes
```

29.21.2 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

29.21.3 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
-------------------------------------------------------------------------------
```

29.21.4 See also

ip set rip hostroutes on page 664

ip set interface rip accept on page 657

ip set interface rip send on page 661

29.22 ip list routes

29.22.1 Syntax

```
ip list routes
```

29.22.2 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)

29.22.3 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
--------------------------------------------------------------------------------
```

29.22.4 See also

ip show route on page 674

29.23 ip ping

29.23.1 Syntax

```
ip ping <dest-ip>
```

29.23.2 Description

This command pings a specified destination IP address.

Note - You can only ping IP addresses. You can **not** ping host names using DNS client.

29.23.3 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
dest-ip	The IP address of the destination machine that you want to ping, displayed in the following format: 192.168.102.3	N/A

29.23.4 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.
```

29.24 ip set interface ipaddress

29.24.1 Syntax

```
ip set interface {<name>|<number>} ipaddress <ipaddress> [<netmask>]
```

29.24.2 Description

This command sets the IP address for an existing IP interface.

29.24.3 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

Option	Description	Default value
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.

29.24.4 Example

```
prompt> ip set interface ip4 ipaddress 192.168.102.3 255.255.255.0
```

29.24.5 See also

ip set interface mtu on page 655

ip set interface dhcp on page 656

ip list interfaces on page 648

ip set interface netmask on page 654

29.25 ip set interface netmask

29.25.1 Syntax

```
ip set interface {<name>|<number>} netmask <netmask>
```

29.25.2 Description

This command sets the netmask for an existing IP interface.

29.25.3 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
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

29.25.4 Example

```
prompt> ip set interface ip6 netmask 255.255.255.0
```

29.25.5 See also

ip set interface ipaddress on page 652

ip list interfaces on page 648

29.26 ip set interface mtu

29.26.1 Syntax

```
ip set interface {<name>|<number>} mtu <mtu>
```

29.26.2 Description

This command sets the MTU (Maximum Transmission Unit) for an existing IP interface.

29.26.3 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*.	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

29.26.4 Example

```
prompt> ip set interface ip2 mtu 800
```

29.26.5 See also

ip set interface ipaddress on page 652

ip set interface dhcp on page 656

ip list interfaces on page 648

29.27 ip set interface dhcp

29.27.1 Syntax

```
ip set interface {<name>|<number>} dhcp {enabled|disabled}
```

29.27.2 Description

This command specifies whether a named interface should obtain its configuration via DHCP.

29.27.3 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
enabled	The interface obtains its configuration information from DHCP client.	disabled
disabled	The interface does not use DHCP client configuration information.	

29.27.4 Example

```
prompt> ip set interface ip2 dhcp enabled
```

29.27.5 See also

ip set interface ipaddress on page 652

ip set interface mtu on page 655

For information on setting DHCP client configuration options, see *DHCP Client CLI commands* on page 61.

ip list interfaces on page 648

29.28 ip set interface rip accept

29.28.1 Syntax

```
ip set interface {<name>|<number>} rip accept
{none|v1|v2|all}
```

29.28.2 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.

29.28.3 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 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.	

29.28.4 Example

```
prompt> ip set interface ip3 rip accept none
```

29.28.5 See also

ip set interface rip send on page 661

ip set interface rip multicast on page 659

ip set rip hostroutes on page 664

ip set rip poison on page 665

ip show on page 671

ip list interfaces on page 648

29.29 ip set interface rip multicast

29.29.1 Syntax

```
ip set interface {<name>|<number>} rip multicast {enabled | disabled}
```

29.29.2 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.

29.29.3 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
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.	

29.29.4 Example

```
prompt> ip set interface ip1 rip multicast enabled
```

29.29.5 See also

ip list interfaces on page 648

ip set interface rip send on page 661

29.30 ip set interface rip send

29.30.1 Syntax

```
ip set interface {<name>|<number>} rip send {none|v1|v2|all}
```

29.30.2 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.

Note - 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.

29.30.3 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

Option	Description	Default value
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.	

29.30.4 Example

```
prompt> ip set interface ip1 rip send v1
```

29.30.5 See also

ip set interface rip accept on page 657

ip set rip hostroutes on page 664

ip set rip poison on page 665

ip show on page 671

ip list interfaces on page 648

For information on RFC1058 and RFC1723, see *http://www.ietf.org/rfc/rfc1723.txt*

29.31 ip set interface tcpmssclamp

29.31.1 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.

29.31.2 Options

The following table gives 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.	

29.31.3 Example

```
prompt> ip set interface ip2 tcpmssclamp enabled
```

29.31.4 See also

ip set interface mtu on page 655

ip show on page 671

IP Stack Programmer's Guide: DO-010017-TC

29.32 ip set rip hostroutes

29.32.1 Syntax

```
ip set rip hostroutes {enabled | disabled}
```

29.32.2 Description

Specifies whether IP interfaces will accept RIP routes to specific routes.

Note - 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.

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
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.	

29.32.4 Example

```
prompt> ip set rip hostroutes enabled
```

29.32.5 See also

ip set interface rip accept on page 657

ip set interface rip send on page 661

ip show on page 671

29.33 ip set rip poison

29.33.1 Syntax

```
ip set rip poison {enabled | disabled}
```

29.33.2 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.

29.33.3 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
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*.	

29.33.4 Example

```
prompt> ip set rip poison enabled
```

29.33.5 See also

ip set interface rip accept on page 657

ip set interface rip send on page 661

ip set rip hostroutes on page 664

ip show on page 671

29.34 ip set route destination

29.34.1 Syntax

```
ip set route {<name>|<number>} destination <dest-network> <netmask>
```

29.34.2 Description

This command sets the destination network address of a route previously created using the *ip add route* command.

29.34.3 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 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

29.34.4 Example

```
prompt> ip set route route1 destination 192.168.103.3 255.255.255.0
```

29.34.5 See also

ip set route gateway on page 667

ip set route cost on page 669

ip list routes on page 650

29.35 ip set route gateway

29.35.1 Syntax

```
ip set route {<name>|<number>} gateway <gateway>
```

29.35.2 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.

29.35.3 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 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

29.35.4 Example

```
prompt> ip set route route1 gateway 192.168.102.3
```

29.35.5 See also

ip add route on page 626

ip set route destination on page 666

ip set route cost on page 669

ip list routes on page 650

29.36 ip set route cost

29.36.1 Syntax

```
ip set route {<name>|<number>} cost <cost>
```

29.36.2 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.

29.36.3 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 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

29.36.4 Example

```
prompt> ip set route route1 cost 3
```

29.36.5 See also

ip add route on page 626

ip set route destination on page 666

ip set route gateway on page 667

ip list routes on page 650

29.37 ip set route interface

29.37.1 Syntax

```
ip set route {<name>|<number>} interface {<interface>|none}
```

29.37.2 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.

29.37.3 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 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

29.37.4 Example

```
prompt> ip set route r1 interface eth1
```

29.37.5 See also

ip list interfaces on page 648

ip list routes on page 650

29.38 ip show

29.38.1 Syntax

```
ip show
```

29.38.2 Description

Shows current RIP configuration and any other information global to the router.

29.38.3 Example

```
prompt> ip show
Global IP configuration:


     Host routes: true
  Poison reverse: false
```

29.38.4 See also

ip set rip hostroutes on page 664

ip set rip poison on page 665

29.39 ip show interface

29.39.1 Syntax

```
ip show interface {<name>|<number>}
```

29.39.2 Description

This command displays the following information about a named interface:

- IP address and netmask address (if set)
- MTU (Maximum Transmission Unit)
- Status of DHCP
- Status of TCP MSS Clamp
- Status of RIP send and RIP accept
- Status of RIP multicast

29.39.3 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

29.39.4 Example

```
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
```

29.39.5 See also

ip show on page 671

ip show route on page 674

ip list interfaces on page 648

29.40 ip show route

29.40.1 Syntax

```
ip show route {<name>|<number>}
```

29.40.2 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

29.40.3 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 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

29.40.4 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:
```

29.40.5 See also

ip show on page 671

ip list routes on page 650

30.TFTP CLI commands

This chapter describes CLI support for TFTP.

30.1 Summary

30.1.1 TFTP CLI commands

There is currently no support for TFTP in the CLI.

30.1.2 TFTP Console commands

The table below lists the *tftp* **console** commands and, if they are usable or blacklisted:

Command	CLI status
connect	Blacklisted command, see *connect* on page 892
get	Blacklisted command, see *get* on page 894
help	Blacklisted command, see *help* on page 895
init	Blacklisted command, see *init* on page 896
list	Blacklisted command, see *list* on page 897
put	Blacklisted command, see *put* on page 898
trace	Blacklisted command, see *trace* on page 899
version	Blacklisted command, see *version* on page 900

31. Transports CLI commands

This chapter describes the Transports CLI commands

31.1 Summary

This module allows you to clear, delete, list and display information about existing transports that were created using the *<transport_module> add transport* commands. To carry out more detailed configuration of transports, see the corresponding transport module chapter:

For RFC1483 commands, see *RFC1483 CLI commands* on page 517

For PPPoA commands, see *PPPoA CLI commands* on page 311

For PPPoE commands, see *PPPoE CLI commands* on page 383

For PPPoH commands, see *PPPoH CLI commands* on page 451

For Ethernet commands, see *Ethernet CLI commands* on page 163

For Frame Relay commands, see *Frame Relay CLI commands* on page 223

For IPoA commands, see *IPoA CLI commands* on page 245

31.1.1 Transports CLI commands

The table below lists the Transports commands provided by the CLI:

Command	Reference
transports clear	*transports clear* on page 681
transports delete	*transports delete* on page 682
transports list	*transports list* on page 683
transports show	*transports show* on page 684

31.2 transports clear

31.2.1 Syntax

```
transports clear
```

31.2.2 Description

This command deletes all transports that were created using the <*transport_module*> *add transport* command.

31.2.3 Example

```
prompt> transports clear
```

31.2.4 See also

transports delete on page 682

31.3 transports delete

31.3.1 Syntax

```
transports delete {<name>|<number>}
```

31.3.2 Description

This command deletes a single transport that was created using the *<transport_module> add transport* command.

31.3.3 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

31.3.4 Example

```
prompt> transports delete eth1
```

31.3.5 See also

transports clear on page 681

transports list on page 683

31.4 transports list

31.4.1 Syntax

```
transports list
```

31.4.2 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

31.4.3 Example

```
prompt> transports list
Services:

 ID  |    Name      | Type
-----|--------------|-----------------------------------------------------
   1 | rfc1483      | RFC1483   | TxPkts:    0/0    RxPkts:    0/0
   2 | pppoh2       | PPP       | TxPkts:    0/0    RxPkts:    0/0
   3 | pppoh1       | PPP       | TxPkts:    0/0    RxPkts:    0/0
   4 | pppoa2       | PPP       | TxPkts:    0/0    RxPkts:    0/0
   5 | eth0         | Ethernet  | TxPkts:    0/0    RxPkts:    0/0
-------------------------------------------------------------------------
```

31.4.4 See also

transports show on page 684

31.5 transports show

31.5.1 Syntax

```
transports show {<name>|<number>}
```

31.5.2 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.

31.5.3 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 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

31.5.4 Examples

Example One - PPP transport

```
prompt> transports show pppoa1
PPP Status

PPP
Summary              : disabled
Server               : true
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     :
Dialout Auth         : none

Interface ID         : 2
Magic Number         : 0
MRU                  : 0
SVC                  : false
Remote Atm           :
Ip Addr From IPCP    : true
Give DNSto Relay     : true
Give DNSto Client    : true
Lcp Echo Every       : 10
If In Octets         : 0
If Out Octets        : 0
If In Errors         : 0
If Out Errors        : 0
Packets Sent         : 0
Good Packets Received : 0
Enabled              : false

Termination          :


Hdlc Channel
Port                 : hdlc
```

Example Two - RFC1483 transport

```
prompt> transports show myrfc1483
RFC1483 Status

RFC1483
Mode                 : LlcBridged
If In Octets         : 0
If Out Octets        : 0
If In Errors         : 0
```

```
If Out Errors        : 0
Packets Sent         : 0
Good Packets Received : 0
Enabled              : true


Atm Channel
Tx Vci               : 600
Rx Vci               : 600

Peak Cell Rate       : 2000
Class                : UBR
Port                 : a1
```

31.5.5 See also

transports list on page 683

32. User CLI commands

This chapter describes the User CLI commands.

32.1 Summary

32.1.1 User CLI commands

The table below lists the *User* commands provided by the CLI:

Command	Reference
user logout	*user logout* on page 689
user password	*user password* on page 690
user change <name>	*user change* on page 691

32.2 user logout

32.2.1 Syntax

```
user logout
```

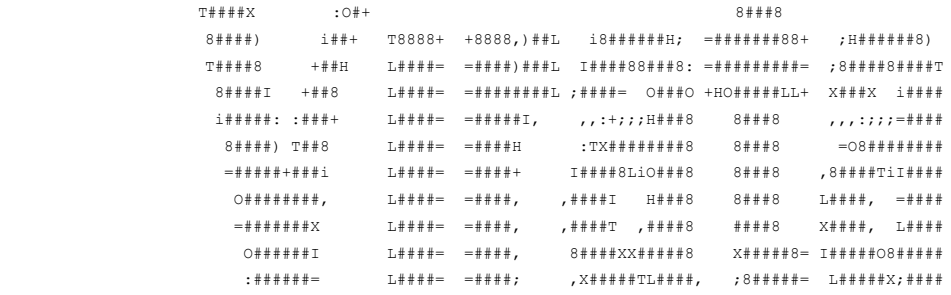
32.2.2 Description

This command logs you out of the system. Default, Engineer and Super users can use this command.

32.2.3 Example

```
prompt> user logout

Logging out.
```



```
Login:
```

32.3 user password

32.3.1 Syntax

```
user password
```

32.3.2 Description

This command allows you to change your user password. Default, Engineer and Super users can use this command.

32.3.3 Example

```
prompt> user password
Enter new password *****
        Again to verify *****
```

32.4 user change

32.4.1 Syntax

```
user change <name>
```

32.4.2 Description

This command allows you to change your login to that of another named user. Super users can use this command. When you change your login to that of a user with Default or Engineer access permissions, you lose your Super user privileges and inherit the access permissions of either the Default or Engineer user.

32.4.3 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 unique login name made up of more than one character that identifies an individual user and lets the user access the system.	N/A

32.4.4 Example

```
prompt> user change admin
You are now logged in as user 'admin' ...
```

32.4.5 See also

system add user on page 591

33. Web Server CLI commands

This chapter describes the Web Server CLI commands.

33.1 Summary

33.1.1 Web Server CLI commands

The table below lists the *Web Server* commands provided by the CLI:

Command	Description/Console command
webserver clear stats	*webserver clear stats* on page 696
webserver enable|disable	*webserver enable|disable* on page 697
webserver set interface	*webserver set interface* on page 698
webserver set managementip	*webserver set managementip* on page 699
webserver set port	*webserver set port* on page 700
webserver set upnpport	*webserver set upnpport* on page 701
webserver show info	*webserver show info* on page 702
webserver show stats	*webserver show stats* on page 703

33.1.2 Web Server Console commands

The table below lists the *Web Server* **console** commands and, if available, their CLI equivalent command:

Command	CLI Equivalent
webserver archive	Usable command see *webserver archive* on page 902
webserver clearstats	Replaced by CLI command *webserver clear stats* on page 696.
webserver {enable|disable}	Replaced by CLI command *webserver enable|disable* on page 697
webserver load	Usable command see *webserver archive* on page 902
webserver mgmtip	Replaced by CLI command *webserver set managementip* on page 699.
webserver port	Replaced by CLI command *webserver set port* on page 700.

Command	CLI Equivalent
webserver stats	Replaced by CLI command *webserver show stats* on page 703.
webserver status	Replaced by CLI command *webserver show stats* on page 703.
webserver version	Replaced by CLI command *webserver show stats* on page 703.

33.2 webserver clear stats

33.2.1 Syntax

```
webserver clear stats
```

33.2.2 Description

This command sets all of the Web Server process counters to 0.

33.2.3 Example

```
prompt> webserver clear stats
```

33.2.4 See also

webserver show info on page 702.

33.3 webserver enable|disable

33.3.1 Syntax

```
webserver {enable|disable}
```

33.3.2 Description

This command enables or disables the Web Server process.

By default, the Web Server process is enabled.

33.3.3 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
enable	Enables the Web Server process.	enable
disable	Disables the Web Server process.	

33.3.4 Example

```
prompt> webserver disable
WebServer is disabled
```

33.4 webserver set interface

33.4.1 Syntax

```
webserver set interface <interface>
```

33.4.2 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* on page 598) and restart your system (see *system restart* on page 611) to activate the Web Server settings.

33.4.3 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
interface	A name that identifies an existing IP interface. To display interface names, use the *ip list interfaces* command.	iplan

33.4.4 Example

```
prompt> webserver set interface ip
```

33.4.5 See also

webserver set upnpport on page 701

For more information on UPnP, see *The ISOS UPnP Internet Gateway Device: DO-009103-TC*.

33.5 webserver set managementip

33.5.1 Syntax

```
webserver set managementip {ip-address}
```

33.5.2 Description

This command allows connection requests to be restricted to only one IP address, (e.g. from an IP address that is used by a management entity) or from any IP address (by setting the IP address to 0.0.0.0).

33.5.3 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
ip-address	The only IP address that the Web Server will allow connection requests from. The IP address is displayed in the following format: 192.168.102.3	0.0.0.0

33.5.4 Example

prompt> **webserver set managementip 192.168.102.3**

```
Management IP address is 192.168.102.3
```

33.6 webserver set port

33.6.1 Syntax

```
webserver set port <port>
```

33.6.2 Description

This command sets the HTTP port number that the Web Server process will use to transfer data.

33.6.3 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
port	A valid port number that must be between 0 and 65535.	80

33.6.4 Example

```
prompt> webserver set port 100
HTTP port number is 100
```

33.6.5 See also

webserver set upnpport on page 701

33.7 webserver set upnpport

33.7.1 Syntax

```
webserver set upnpport <port>
```

33.7.2 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* on page 598) and restart your system (see *system restart* on page 611) to activate the Web Server settings.

33.7.3 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
port	A valid UPnP port number that must be between 0 and 65535.	N/A

33.7.4 Example

```
prompt> webserver set upnpport 280
```

33.7.5 See also

webserver set port on page 700

For more information on UPnP, see *The ISOS UPnP Internet Gateway Device: DO-009103-TC*.

33.8 webserver show info

33.8.1 Syntax

```
webserver show info
```

33.8.2 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

33.8.3 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
```

33.8.4 See also

webserver clear stats on page 696.

33.9 webserver show stats

33.9.1 Syntax

```
webserver show stats
```

33.9.2 Description

This command tells you how many bytes have been transmitted and received by the Web Server.

33.9.3 Example

```
prompt> webserver show stats


Web Server statistics:
Bytes transmitted:       2122
Bytes received:          0
```

33.9.4 See also

webserver show info on page 702

A:ATMOS Console Commands

This chapter describes the ATMOS Console commands.

A.1 General notes

Example output is shown only to clarify the description of the commands; the actual output is not necessarily in exactly the same format.

A.2 event …

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

A.2.1 CLI access permission

Users with the following access permissions can use this command:

- Engineer
- Super user

A.2.2 Syntax

```
event help

event n[ext]

event p[revious]

event r[ecent]

event show

event unshow
```

A.2.3 Description

The command *event show* enables display of background output on this console device.

The command *event unshow* disables it. By default, the display of background output is disabled.

The command *event recent* (or *event r*) displays the most recent background output stored in the memory buffer; *event previous* (or *event p*) displays the background output immediately preceding that last displayed; *event next* (or *event n*) displays the background output immediately following that last displayed. Up to 24 lines are displayed in each case.

For example, after *event r*, *event n* will show only new background output that has arrived since the *event r* command: repeated typing of *event n* will let the user keep up to date with new background output (without any repetitions in the output).

The command *event help* displays a summary of the options of the *event* command.

A.3 restart

Note - This console command **has** been replaced by a CLI command. See *system restart* on page 611. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

A.3.1 Syntax

```
restart
```

A.3.2 Description

Reboots the ATMOS system.

The reboot will cause the loss of any configuration which has not yet been saved (using the *config save* command).

A.3.3 See also

tell <process> ... on page 722.

A.4 uptime

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

A.4.1 CLI access permission

Users with the following access permissions can use this command:

- Super user

A.4.2 Syntax

```
uptime
```

A.4.3 Description

Displays the time for which the system has been ‘up’, i.e. the time since the system has been powered up or restarted (using the *restart* command).

A.4.4 See also

restart on page 709.

A.5 version

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

A.5.1 CLI access permission

Users with the following access permissions can use this command:

- Engineer
- Super user

A.5.2 Syntax

```
version
```

A.5.3 Description

Displays the system type and version.

A.6 <process>, <process> <command>

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

A.6.1 CLI access permission

Users with the following access permissions can use this command:

- Super user

A.6.2 Syntax

```
<process> <command>

<process>

home

home <command>
```

A.6.3 Description

In these commands, <*process*> can be any of a list of process names known to the console.

The former variant sends the command as a TELL message to the process.

The latter variant remembers the process name, and sends subsequent commands as TELL messages to the process, as if they had been preceded by the process name, until the command *home* is issued. The prompt is changed to reflect this; moreover, if a *help* command with no arguments is issued, it is passed to the process as usual, but then information about the *home* command is appended to the process's output by the console.

A.6.4 Example

```
prompt>isfs version
ISFS v2.07
prompt>isfs
prompt isfs> version
ISFS v2.07
prompt isfs> help
ISFS commands are:
help        - this text is displayed
ls          - list ISFS files
rm <file>   - remove file from ISFS
cat <file>  - show file contents
version     - displays version number

Use "home" to return to "prompt>"

prompt isfs> home
prompt>
```

When the console is at the prompt of a particular process, the command *home* <*command*> or *home* <*process*> <*command*> may be used to execute a command as if the user had typed *home* followed by <*command*> or <*process*> <*command*>. However, the console will remain at the same process prompt.

The command *home* <*process*> will change the prompt from the current process to a new process <*process*>.

A.6.5 Example

```
prompt> bridge

prompt bridge> version

Bridge Version 1.15

prompt bridge> home version

Ethernet Gateway Version 7.0.0.7 (2 Jun 2000)

prompt bridge> home nat version

NAT Version 2.02

prompt bridge> home edd

prompt edd> version

EDD Version 1.03

prompt edd> home

prompt>
```

A.6.6 See also

tell <process> ... on page 722.

A.7 . (history mechanism)

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

A.7.1 CLI access permission

Users with the following access permissions can use this command:

- Engineer
- Super user

A.7.2 Syntax

.

A.7.3 Description

Repeats the previous console command.

A.7.4 Example

```
prompt> isfs version
ISFS v2.07
prompt> .
ISFS v2.07
```

A.8 @ commands

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

A.8.1 CLI access permission

Users with the following access permissions can use this command:

- Super user

A.8.2 Syntax

```
@@<line>

@ <line>

@<process> <line>

@<process>
```

A.8.3 Description

Lines beginning with the @ character are intercepted by the console even when the console device is bound to a file handle.

To bypass this interception and pass a line beginning with @ to a process, the @ must be doubled; the line with one @ removed will be passed on like a normal input line.

If the @ is followed by a space (or any non-alphanumeric character), the remainder of the line is treated as a console command, as if the device were not bound.

The @<*process*> <*line*> form passes <*line*> to a file (if any) opened for reading by the named process.

The @<*process*> form binds the console device to the named process, in the same way as *bind* <*process*>. (Except that the latter, not being an @ command, will not work if the console device is bound. More generally, @<*process*> does the same as @*bind* <*process*>.)

A.8.4 Example

```
prompt> @ip
```

(The *ip>* prompt does not appear until the *Enter* key is pressed again.)

```
ip> device
# type          dev file                IP address
device ether    ether  //edd mtu 1500    192.168.3.55
ip> @console
prompt>
```

A.8.5 See also

bind <process>, unbind on page 726.

A.9 Special-purpose commands

This section lists commands that are normally useful only to developers rather than to normal users, or else are retained only for consistency with older versions of the software. They are not described in the output of the *help* command.

A.10 list

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

A.10.1 CLI access permission

Users with the following access permissions can use this command:

- Engineer
- Super user

A.10.2 Syntax

```
list
```

A.10.3 Description

The *list* command lists the active console devices (referred to as *threads*) and files.

For each console device, if it is bound to a file then the list shows which file it is bound to; if background output is enabled on that device (see *event ...* on page 707) then the list indicates the fact.

For each file, the list shows the name of the process that opened the file and the number of read commands outstanding on the file. If the file is bound to a device then the list shows which device it is bound to; if the file is for foreground output then the list indicates the fact (with the string *FG*).

A.10.4 Example

```
prompt> list
Threads:
  1: ACTIVE, FP 00730520
  3: ACTIVE, FP 00719170, Bound 75, events shown

Files:
  0: OPEN FP 00718e70, Queue chips, 0 read(s)
  1: OPEN FP 00718c30, Queue isfs, 0 read(s)
```

(some output omitted)

```
 49: OPEN FP 00715af4, Queue ip, 0 read(s), Bound 3, FG
(some output omitted)
 75: OPEN FP 00715b38, Queue ip, 1 read(s), Bound 3
```

(some output omitted)

A.11 echo …

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

A.11.1 CLI access permission

Users with the following access permissions can use this command:

- Engineer
- Super user

A.11.2 Syntax

```
echo <text>
```

A.11.3 Description

Echoes the text. (Not a very useful command.)

A.11.4 Example

```
prompt> echo hello world
hello world
```

A.12 tell <process> …

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

A.12.1 CLI access permission

Users with the following access permissions can use this command:

- Super user

A.12.2 Syntax

```
tell <process> <command>
```

A.12.3 Description

Sends the command as a TELL message to a specific process. Note that for many processes the *tell* can be omitted; see "*<process>, <process> <command>* on page 712".

A.12.4 Example

```
prompt> tell hswctrl portinfo a1
port  type vers  flags
A1  25Mbps 1QUA  mast uni30 ilmi netside tx8khz manconfig
```

A.12.5 See also

<process>, <process> <command> on page 712.

A.13 exit, exit!

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

A.13.1 CLI access permission

Users with the following access permissions can use this command:

- Super user

A.13.2 Syntax

```
exit
```

```
exit!
```

A.13.3 Description

Exits from ATMOS to the boot ROM. Without the exclamation mark, the command works only from the serial interface; with the exclamation mark it works from any console device.

Note - This command is now deprecated and provides no useful output.

A.14 debug

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

A.14.1 CLI access permission

Users with the following access permissions can use this command:

- Super user

A.14.2 Syntax

```
debug
```

A.14.3 Description

Enters the ATMOS debugger. Only works when issued at the serial interface. (Since the ATMOS debugger talks to the serial interface, the *debug* command would be of little use elsewhere.)

A.15 crlf, nocrlf

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

A.15.1 CLI access permission

Users with the following access permissions can use this command:

- Super user

A.15.2 Syntax

```
crlf
```

```
nocrlf
```

A.15.3 Description

Controls whether line-feed characters written to this console device are output as carriage-return/line-feed pairs (*crlf*) or just as single line-feed characters (*nocrlf*).

A.16 bind <process>, unbind

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

A.16.1 CLI access permission

Users with the following access permissions can use this command:

- Super user

A.16.2 Syntax

```
bind <process>
```

```
unbind
```

A.16.3 Description

The former command binds this console device to the specified process – that is, binds this device to the file, if any, opened for read by that process, and binds every file opened by the process to this device.

The latter command *unbinds* this console device – that is, undoes the above bindings.

A.16.4 Example

```
prompt> bind ip
```

```
ip> @ unbind
```

```
prompt>
```

A.16.5 See also

@ commands on page 716.

A.17 buildid

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

A.17.1 CLI access permission

Users with the following access permissions can use this command:

- Engineer
- Super user

A.17.2 Syntax

```
buildid
```

A.17.3 Description

This command displays the build ID - a unique identifier for the image that you have built. If you need to contact the GlobespanVirata Technical Advice Center regarding this build, you must give them the correct build ID.

A.17.4 Example

```
prompt> buildid
Build ID: BOGDOV-DZWEUQ-FYDGEH
```

B: Bridge Console Commands

This chapter describes the Bridge Console commands.

B.1 device add

Note - This console command **has** been replaced by the CLI command *bridge add interface* on page 33. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

B.1.1 Syntax

```
device add <device>
```

B.1.2 Description

This command adds a device to the bridge configuration. Attempts to add the bridge itself or an existing device to the bridge are rejected. Attempts to add devices which don't support the Cyan interface are rejected. There is a limit on the number of devices that can be attached to the bridge. If the device being added is from a process which supports multiple devices, the */DEVICE* attribute must be specified as part of the device name. The table below shows devices which may be attached to the bridge, although not all systems may support all devices.

Device	Remarks	Source
edd	Ethernet driver	bun_ethernet
bun	RFC1483 protocol (PVC)	rfc1483
ppp	Point-to-Point protocol	pp

Configuration saving saves this information.

See the section entitled *Implementation Constraints* in the *ATMOS Transparent Bridge Specification, DO-007087-PS*, for details of which devices are added by default.

B.1.3 Example

Simple examples

```
device add edd
```

```
device add ppp/DEVICE=2
```

Using the BUN RFC1483 driver

This example shows how to add the BUN RFC1483 driver, dynamically from the console. You need to define and configure a device and a port.

Normally, the RFC1483 BUN device will pass all data straight through, untouched. This means that even though you have changed your port definition to include the RFC1483 driver, you can still use other protocols on the same port. In order to enable RFC1483 encapsulation, the RFC1483 attribute on the channel must be set to *true*.

The channel attribute *mode* dictates the functional behaviour of the driver, in terms of encapsulation method to use and traffic nature (bridged/routed). The channel attribute *promiscuous* selects the promiscuity behaviour of the driver.

The driver requires, at configuration time, to be layered with the BUN utopia and nec98408 devices, in order to be used. So, for the sake of the following examples, let's assume that the related BUN port is called *rfc_port*, and it has been configured in the following way:

```
device: rfc_dev = rfc1483, nec98408, utopia

port : rfc_port = rfc_dev/PhysicalPort=0/PortSpeed=59111
```

If we want to attach the device to the bridge, then the following command must be issued (all typed on one line):

```
bridge device add //bun/port=rfc_port/rfc1483=true

/mode=llcbridged/txvci=600/rxvci=600
```

The above command creates a channel with RFC1483 enabled, and it uses the LLC encapsulation for bridged traffic. The next command, is the same, however it uses the VC multiplexing method:

```
<all typed in one line>

bridge device add //bun/port=rfc_port/rfc1483=true

/mode=vcmuxbridged/txvci=600/rxvci=600
```

B.1.4 See also

device delete on page 732 and *device list* on page 733.

B.2 device delete

Note - This console command **has** been replaced by the CLI command *bridge delete interface* on page 36. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

B.2.1 Syntax

```
device delete <device>
```

B.2.2 Description

This command deletes a device from the bridge configuration. The syntax of the device name is the same as that for the *device add* command.

Configuration saving saves this information.

B.2.3 Example

```
device delete r1483
```

B.2.4 See also

device add on page 730 and *device list* on page 733.

B.3 device list

Note - This console command **has** been replaced by the CLI command *bridge list interfaces* on page 38. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

B.3.1 Syntax

```
device list
```

B.3.2 Description

This command lists all the devices that are currently attached to the bridge. It does not show the stored configuration (which can be seen with the *config print* command).

B.3.3 Example

```
device list
```

B.3.4 See also

device add on page 730 and *device delete* on page 732.

B.4 ethertype

Note This console command **has** been replaced by the CLI command *bridge set interface filtertype* on page 40. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

B.4.1 Syntax

```
ethertype [<port> any|ip|pppoe]
```

B.4.2 Description

This command enables filtering of Ethernet packets according to the ETHER_TYPE field in the header. Only packets of the type specified using this command will be **sent** on the port specified; packets of all types will always be **received**.

By default, all bridge ports are set to *any*, which means that the type of the packet will never be checked. The meaning of the other options is as follows:

Option	Permitted ETHER_TYPE values
ip	0x0800 - IP 0x0806 - ARP
pppoe	0x8863, 0x8864 - PPP Over Ethernet (RFC 2516)

The port is specified as an integer, as displayed by the device list command. When using this command in the *initbridge* configuration file, ports are numbered in the order in which the device add commands are given, starting from 1.

If no arguments are given, the current settings for each port are displayed.

B.4.3 Example

```
ethertype 2 any
```

B.5 filter

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

B.5.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

B.5.2 Syntax

```
filter
```

B.5.3 Description

This command shows the current contents of the bridge's filter table. The MAC entries for each device are shown in turn together with the time that the MAC address was last seen by the bridge. The command also shows the current filter ageing time, in seconds, and the number of creation failures since the system was started. Creation failures occur when there is no room left in the filter table for a new entry.

B.5.4 Example

```
filter
```

B.5.5 See also

filterage on page 736.

B.6 filterage

Note This console command **has** been replaced by the CLI command *bridge set filterage* on page 39. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

B.6.1 Syntax

```
filterage [<age>]
```

B.6.2 Description

This command sets, or displays if no arguments are given, the filter table ageing time. The ageing time is the time after which MAC addresses are removed from the filter table when there has been no activity. The time is specified in seconds and may be any integer value in the range 10…100,000 seconds. This value may also be changed through SNMP. Changing the value of *filterage* has immediate effect.

Configuration saving saves this information. By default, the filter ageing time is set to 300 seconds.

B.6.3 Example

```
filterage
```

B.6.4 See also

filter on page 735.

B.7 flush

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

B.7.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

B.7.2 Syntax

```
flush [<port>]
```

B.7.3 Description

This command allows the MAC entries for a specified port, or all ports, to be removed from the filter table. The port number for a device may be determined using the *device list* or *status* commands. If the port number is omitted, all entries for all ports are removed from the filter table.

B.7.4 Example

```
flush
```

B.7.5 See also

filter on page 735, *device list* on page 733, *status* on page 745.

B.8 info

Note - This console command **has** been replaced by the CLI command *bridge show* on page 47. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

B.8.1 Syntax

```
info
```

B.8.2 Description

This command displays build information about the **bridge** process. The *version* command is a synonym for this command.

B.8.3 Example

```
info
```

B.8.4 See also

version on page 746.

B.9 interface

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

B.9.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

B.9.2 Syntax

```
interface [sub-command]
```

B.9.3 Description

This command accesses the ethernet support library sub-commands for the bridge itself, not for the devices which are attached to it.

The ethernet support commands are documented in the *ATMOS Ethernet Support Library Specification, DO-007502-PS*.

B.9.4 Example

```
interface stats
```

B.10 portfilter

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

B.10.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

B.10.2 Syntax

```
portfilter [<source port> all|<destination ports>]
```

B.10.3 Description

The portfilter command allows control over the bridge's forwarding and broadcasting behaviour. By default, when a multicast or an unknown packet is received on a port (referred to above as the source port), it will be forwarded to all other bridge ports (referred to above as the destination ports).

Each bridge port may have its behaviour modified separately. The first example below configures the bridge so that packets arriving on port 2 will only be forwarded to ports 3, 4 and 5, and packets arriving on port 3 will only be forwarded to port 1. All other ports retain their default behaviour.

Note that this command does not force packets arriving on the source port to be sent to all specified destination ports. The bridge retains its learning behaviour, so unicast packets, once their destination is known to the bridge, will still only be sent to one port. Note also that the bridge itself (for example when attached to the IP router) will always forward to all ports, and will always be forwarded to by all ports.

The default behaviour can be restored by calling this command with the argument *all*, as shown in the second example.

The ports are specified as integers, as displayed by the *device list* command. When using this command in the initbridge configuration file, ports are numbered in the order in which the *device add* commands are given, starting from 1.

If no arguments are given, the current settings for each port are displayed.

B.10.4 Example 1

```
portfilter 2 3 4 5
portfilter 3 1
```

B.10.5 Example 2:

```
portfilter 2 all
portfilter 3 all
```

B.10.6 See also

device add on page 730 and *device list* on page 733.

B.11 spanning

Note - The following console commands **have** been replaced by the CLI command *bridge set spanning* on page 41:

```
spanning enable/disable
spanning forwarddelay
spanning hellotime
spanning info
spanning maxage
spanning priority
spanning status
spanning version
```

Use the CLI command instead of the console commands. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

Note - The console command *bridge spanning event* **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

B.11.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

Note - The console command *bridge spanning port* **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

B.11.2 CLI access permission - usable commands

Users with the following access permissions can use the blacklisted commands:

- Super user

B.11.3 Syntax

```
spanning [sub-command]
```

B.11.4 Description

The spanning tree commands are only available if it has been compiled in to the bridge.

The following spanning tree sub-commands are available:

Sub-command	Description
disable	Disables the spanning tree process
enable	Enables spanning tree process
event [<level>]	Sets the level of event reporting
forwarddelay [<time>]	Reads or sets the time (in seconds) in which the bridge remains in the listening or learning states
hellotime [<time>]	Reads or sets the time (in seconds) after which the spanning tree process sends notification of topology changes to the root bridge
info	Displays the version number of the spanning tree implementation
maxage [<time>]	Reads or sets the maximum age of received spanning tree protocol information before it is discarded
port <number>	Controls the configuration of the bridge's ports as far as the operation of the spanning tree protocol is concerned. Options are: port <number> disable: disables a port port <number> enable: enables a port port <number> pathcost: reads or sets the cost of using this port port <number> priority: Reads or sets the priority of the port
priority [,bridgepriority>]	Reads or sets the priority of the bridge (any value in the range 0 to 65535)
status	Reports the status of the spanning tree
version	Displays the version number of the spanning tree implementation

The spanning tree commands are documented in the *ATMOS Spanning Tree Specification, DO-007085-PS*.

B.12 status

Note - This console command **has** been replaced by the CLI commands *bridge show* on page 47 and *bridge list interfaces* on page 38. Use the CLI commands instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use these CLI commands.

B.12.1 Syntax

```
status
```

B.12.2 Description

This command shows the status of the bridge and its ports. The status information for a port includes the SNMP type information about time-exceeded packets, packets discarded, etc. It also includes the broadcast history of the port over the last five seconds and the *high water mark* of packets queued on the bridge for this device.

B.12.3 Example

```
status
```

B.13 version

Note - This console command **has** been replaced by the CLI command *bridge show* on page 47. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

B.13.1 Syntax

```
version
```

B.13.2 Description

This command displays build information about the *bridge* process. The *info* command is a synonym.

B.13.3 Example

```
version
```

B.13.4 See also

```
info
```

C:BUN Console Commands

This chapter describes the BUN Console commands.

C.1 Introduction

C.1.1 Scope

A description is provided of the use of console commands.

No information on implementing additional commands is given in this chapter: implementers of new BUN devices may provide access to diagnostic or status information by implementing attributes to handle these tasks. The standard BUN console commands may then be used to display or change these settings.

Command parsing is case insensitive. White-space may be used to separate distinct arguments. Any prefix of the string *bun* to the command line is ignored.

C.1.2 Build Inclusion

The full BUN console command set is included with all builds that include the BUN package.

To include the BUN package, add the following directive to the ATMOS SYSTEM file:

```
package bun
```

The directive may be placed anywhere in the SYSTEM file after the inclusion of the core package (*core.pkg*).

C.1.3 Compile Time Configuration

Most BUN commands are available irrespective of the compilation options. This section describes exceptions to this rule.

build

The build command displays the compile-time options, and so will change according to what compilation options are used…

Any compile option that affects BUN operation should be displayed by this command.

C.1.4 Command arguments

devicename

The name of a device.

Device names are either implicit (i.e.: provided from the compiled-in device code) or explicit (i.e.: from a *device:* configuration request).

Device names may contain upper or lower case letters, but use case insensitive matching.

portname

The *name* of a port. This can take several forms:

- The name given on the *port* configuration request
- The alias name specified in the port's *Alias* attribute
- The name as a <*class*>:<*instance*> pair. For example, *atm:0* to reference the first port supporting ATM cell traffic.
- The BUN port number. For example, *0* to refer to the first port.

The last option may be dropped in a future software release.

Port names may contain upper or lower case letters, but all name matching is case insensitive.

channelnumber

The number of a channel. Within a port, each available channel is identified within BUN by a unique channel number. Channel numbers are positive integers, assigned from zero upwards.

To determine the channel numbers that are currently in use, use the *list channels* command to show all active channels on a port (or ports).

Note that to be uniquely specified, both a port name and channel number must be given to console commands which display or manipulate channels.

classname

The name of a class.

Class names may contain upper or lower case letters, though class name matching is always case insensitive.

By default, BUN provides the following class definitions:

- *all*: All ports in the system
- *atm*: All ports supporting ATM cell traffic
- *adsl*: All ports using the ADSL hardware interface
- *ethernet*: All ports using an ethernet hardware interface

- *hdlc*: All ports using an HDLC hardware interface
- *pci*: All ports using a PCI hardware interface
- *usb*: All ports using a USB hardware interface

A running system may contain additional classes specified via the *class* configuration directive (see the commands *list classes* on page 760 and *list config* on page 755).

If necessary, commands may be quoted using angle brackets or double quotes. This prevents the stripping of white-space from the input line. For example:

```
set port atm/usercomment="This is a comment string"
set port atm/usercommand=<An alternative syntax>
```

Within either form of quoted section, the corresponding close quote character may be embedded by prefixing with a backslash. So you could write:

```
set port atm/usercomment=<This is a "cell based" port>
set port atm/usercomment="This is a \"cell based\" port>
```

Mostly you probably won't need to worry about quotation, but be aware of it's effects if you do.

The remainder of this section describes the commands themselves.

C.2 help

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

C.2.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

C.2.2 Syntax

```
help [<command>]
```

C.2.3 Description

Display command information.

If used without the optional command name, a summary of the commands available will be displayed.

If used with a command name, brief usage information will be shown for the command.

Note - Commands listed but which are not covered by this documentation are **not** supported, and may not be present in future software releases.

C.2.4 Examples

```
help
```

```
help set port
```

Note: This command is not intended to replace this documentation, and provides only a very basic level of detail.

C.3 version

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

C.3.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

C.3.2 Syntax

```
version
```

C.3.3 Description

Display the BUN software version.

C.3.4 Example

```
version
```

C.3.5 See also

build on page 753.

C.4 build

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

C.4.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

C.4.2 Syntax

```
build
```

C.4.3 Description

Display information about compile-time build options. For example, if tracing or debug code has been compiled into the image.

C.4.4 Example

```
build
```

C.4.5 See also

version on page 752.

C.5 config

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

C.5.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

C.5.2 Syntax

```
config <configurationstring>
```

C.5.3 Description

Issue a configuration request to BUN.

This command can be used to pass arbitrary configuration strings to BUN, effectively calling *bun_ConfigMakeRequest()* with the supplied configuration string.

This may be used to create new devices or ports at run time, using the same syntax as the configuration strings in the SYSTEM file BUN_CONFIG_<n> directives. This can be particularly useful during the development of new software.

C.5.4 Example

```
config device : nuclear = detonator, uranium
config port : launch = nuclear/silo=3
```

This can also be written as simply:

```
device : nuclear = detonator, uranium
port : launch = nuclear/silo=3
```

C.5.5 See also

list config on page 755.

C.6 list config

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

C.6.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

C.6.2 Syntax

```
list config
```

C.6.3 Description

List the configuration requests that have been passed to BUN.

BUN records all configuration requests that are issued, and maintains information about their parsing. Configuration requests can be in one of three states:

- *Completed* – the request has completed successfully
- *Pending* – the request is stalled, pending creation of a (as yet) non-existent device
- *Failed* – the request failed

Each request is displayed together with any relevant information. In the case of failed requests, an error code is given and the point at which parsing of the configuration string failed is highlighted.

Stalled requests can be unblocked by creating a new device with suitable properties by using the BUN *config* console command to issue a *device* configuration request.

This command is extremely useful for diagnosing problems with device or port configuration.

C.6.4 Example

```
list config
```

C.6.5 See also

config on page 754.

C.7 list devices

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

C.7.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

C.7.2 Syntax

```
list devices
```

C.7.3 Description

List all available devices.

This will show all devices, regardless of how they were created. This includes devices which were compiled into the system (such as the *utopia* device), and compound devices which were created by configuration requests (such as the *atm25* device, a compound of the *utopia* and *nec98408* devices).

C.7.4 Example

```
list devices
```

C.7.5 See also

show device on page 758.

C.8 show device

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

C.8.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

C.8.2 Syntax

```
show device <devicename>
```

C.8.3 Description

Display information about at device.

This displays information about a device in the following format:

Name:<*devicename*>

Description<*devicedescription*>

Contains:<*devicelist*>

The device name is the root name of the device. This is the same as the name passed to the *show device* command.

The device description is a brief string describing the device. For compiled in devices, this string is provided by the driver code. For compound devices, this string is the configuration request used to create the device.

The device list shows which driver code is invoked by this device. For a compiled in device, this will just be the device itself. For a compound device, this will be the list of devices linked to form the compound driver.

C.8.4 Example

```
show device utopia
```

```
show device atm25
```

C.8.5 See also

list devices on page 757.

C.9 list classes

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

C.9.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

C.9.2 Syntax

```
list classes
```

C.9.3 Description

List available port classes on the console. The class name is displayed, together with the necessary attributes for a port to be a member of said class.

C.9.4 Example

```
list classes
```

C.9.5 See also

list classes on page 760.

C.10 show class

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

C.10.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

C.10.2 Syntax

```
class <classname>
```

C.10.3 Description

List members of the specified port class.

C.10.4 Example

```
show class atm
```

C.10.5 See also

list classes on page 760.

C.11 list ports

Note - This console command **has** been replaced by the CLI command *port ?* on page 305. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

C.11.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

C.11.2 Syntax

```
ports
```

C.11.3 Description

List all available ports on the console, in the following format:

<*portnumber*> : <*portname*>

All BUN console which require a port to be identified can accept either the port number or port name as an argument. They may also be used as the argument to a */port=* attribute in *fopen()* `strings`.

C.11.4 Example

```
ports
```

C.11.5 See also

show port on page 763.

C.12 show port

Note - This console command **has** been replaced by the CLI command *port show* on page 308. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

C.12.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

C.12.2 Syntax

```
port <portname>
```

C.12.3 Description

Display detailed information about a port.

This command enumerates all attributes for a port and displays them on the console. It is useful to determine the properties of a port.

C.12.4 Example

```
>bun show port a1
```

C.12.5 See also

list ports on page 762, *set port* on page 764.

C.13 set port

Note - This console command **has** been replaced by the CLI command *port set* on page 306. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

C.13.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

C.13.2 Syntax

```
set port <portname> / <attributelist>
```

C.13.3 Description

Modify a port attribute.

This command may be used to modify an attribute on a port, overriding any values specified in the original port configuration request. The effects of changing any such attributes are device dependent.

This command is intended for development purposes only.

C.13.4 Example

```
set port ethernet /usercomment=”An Ethernet network port”
```

C.13.5 See also

list ports on page 762, *set port* on page 764.

C.14 list channels

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

C.14.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

C.14.2 Syntax

```
list channels [<portname> ]
```

C.14.3Description

List all open connections on the specified port. If no *portname* is specified, all channels on all ports will be displayed.

The channels are shown with their identification number and a selection of *useful* attributes. A full attribute list can be obtained via the *show channel* command.

All channels are shown with the *Enabled* attribute first, which indicates if the channel has yet been enabled (connected) by the application code.

C.14.4 Examples:

```
list channels 0
```

```
list channels atm:0
```

C.14.5 See also

show channel on page 767, *list ports* on page 762, *show port* on page 763.

C.15 list all open channels

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

C.15.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

C.15.2 Syntax

```
list all open channels [<portname> ]
```

C.15.3 Description

This command is similar to the *list channels* command. The *list channels* command shows channels which are either *enabled* or *open*. The *list all open channels* command only shows channels which are *open*.

If no *portname* is specified, all channels on all ports will be displayed.

The channels are shown with their identification number and a selection of *useful* attributes. A full attribute list can be obtained via the *show channel* command.

C.15.4 See also

list channels on page 765.

C.16 show channel

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

C.16.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

C.16.2 Syntax

```
show channel <portname> <channelnumber>
```

C.16.3 Description

Display information about the specified channel. The channel identification number may be obtained from the *list channels* command. All attribute values for the channel are displayed on the console.

Note that you must specify both a port name and channel number. Channel numbers are only unique within a given port.

Also note that, unlike the old ATM driver, the channel number is **not** the same as the receive VCI number.

It is also possible to display channels that are not currently opened by an application. The *bun.active* attribute will return *true* if a channel is currently open, else *false*. Note that a channel handle may be closed and then re-opened by an application at any time – be cautious when using this command.

C.16.4 Example

```
show channel atm 0
```

C.16.5 See also

set channel on page 768, *list channels* on page 765.

C.17 set channel

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

C.17.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

C.17.2 Syntax

```
set channel <portname> <channelnumber> / <attributelist>
```

C.17.3 Description

Modify attributes on the specified channel.

This command allows you to change the attribute values for a given channel. The effect of any changes will be device dependent.

Use this command with extreme caution. The same warnings about an application closing and reopening a channel handle apply as they do for the *show channel* command. Also beware that the application will not be explicitly notified of any changes made, though if it queries its own attribute data it will pick up any changes that have been made.

This command is intended for development purposes only.

C.17.4 Example

```
set channel atm 27 /txvci=32/rxvci=32/pcr=1234
```

C.17.5 See also

list channels on page 765, *show channel* on page 767.

C.18 reset port

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

C.18.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

C.18.2 Syntax

```
reset port <portname>
```

C.18.3 Description

Re-initialise port hardware.

This may be used to request that a device re-initialise the underlying hardware. Not all devices implement this command.

This command is primarily intended for use during test and development of new hardware devices.

C.18.4 Example

```
reset port 3
```

C.18.5 See also

list ports on page 762, *show port* on page 763, *set port* on page 764.

D:Chips Console Commands

This chapter describes the ATMOS Console commands for the chips process.

D.1 cpu

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

D.1.1 CLI access permission

Users with the following access permissions can use this command:

- Engineer
- Super user

D.1.2 Syntax

```
cpu
```

D.1.3 Description

Displays the recent CPU utilization as a percentage. This is a fairly crude measurement: the ATMOS kernel measures the time that the CPU spends in the idle loop over successive three-second intervals, and the *cpu* command uses this measurement from the most recent complete three-second interval.

D.2 debug

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

D.2.1 CLI access permission

Users with the following access permissions can use this command:

- Super user

D.2.2 Syntax

```
debug
```

D.2.3 Description

Enters the ATMOS debugger.

D.3 exit

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

D.3.1 CLI access permission

Users with the following access permissions can use this command:

- Super user

D.3.2 Syntax

```
exit
```

D.3.3 Description

Exits from ATMOS to the boot ROM.

Note - This command is now deprecated and provides no useful output.

D.4 help

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

D.4.1 CLI access permission

Users with the following access permissions can use this command:

- Engineer
- Super user

D.4.2 Syntax

```
help

?

help <command>
```

D.4.3 Description

The *help* command lists all chips commands. *?* is a synonym for this command;

help <*command*> displays more detailed help on the specified command.

This command is available only if the pre-processor symbol *CHIPSHELP* is defined.

D.5 info

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

D.5.1 CLI access permission

Users with the following access permissions can use this command:

- Engineer
- Super user

D.5.2 Syntax

```
info
```

D.5.3 Description

Displays some system type information, version number, and the MAC addresses.

D.5.4 Example

```
> info
word at 0x001C4B54 contains 0x0000337E
```

D.6 mem

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

D.6.1 CLI access permission

Users with the following access permissions can use this command:

- Engineer
- Super user

D.6.2 Syntax

```
mem
```

D.6.3 Description

Displays a summary of how much memory is used by each ATMOS process (distinguishing between heap and thread stacks, along with some other minor categories), along with the amount of free heap memory and the size of the largest single free block.

D.7 rb, rh, rw, wb, wh, ww

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

D.7.1 CLI access permission

Users with the following access permissions can use this command:

- Super user

D.7.2 Syntax

```
rb <addr>
rh <addr>
rw <addr>
wb <addr> <val>
wh <addr> <val>
ww <addr> <val>
```

D.7.3 Description

Reads the byte (*rb*), word (*rw*) or half-word (*rh*) at a specified address, or writes a specified value to the byte (*wb*), word (*ww*) or half-word (*wh*). Addresses and values are specified in hexadecimal, with an optional *0x* prefix.

D.7.4 Example

```
> rw 1c4b54
word at 0x001C4B54 contains 0x0000337E
> rb 1c4b55
byte at 0x001C4B55 contains 0x33
> wb 1c4b56 0x20
value 0x20 written to byte at 0x001C4B56
> rw 1c4b54
word at 0x001C4B54 contains 0x0020337E
> ww 0x1c4b54 14c44
```

```
value 0x00014C44 written to word at 0x001C4B54
```

D.8 steal

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

D.8.1 CLI access permission

Users with the following access permissions can use this command:

- Super user

D.8.2 Syntax

```
steal memory use <handle> <amount>
steal memory release <handle>
steal file use <handle> <device>
steal file release <handle>
steal cpu use <percentage>
steal cpu release
steal status [memory] [file] [cpu]
```

D.8.3 Description

Uses up heap memory, file handles, or CPU cycles. <*handle*> is a number from 0 to 19, used to identify the resource for a later *steal ... release* command.

This command is intended to help test system behaviour when resources are limited, and is available only if the pre-processor symbol *CHIPS_STEAL* is defined.

D.9 tell

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

D.9.1 CLI access permission

Users with the following access permissions can use this command:

- Super user

D.9.2 Syntax

```
tell <process> <command>
```

D.9.3 Description

Sends the command as a TELL message to a specific process. (The same as the console *tell* command.)

E:DHCP-client Console Commands

This chapter describes the DHCP-client Console commands.

E.1 config

Note - This console command **has** been replaced by the CLI command *dhcpclient show* on page 98. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

E.1.1 Syntax

```
dhcpclient config
```

E.1.2 Description

This command displays the current configuration of the DHCP client, including selected DHCP options.

E.1.3 Example

```
prompt> dhcpclient config
---
DHCP client configuration file: '//isfs/dhclient.conf'

timeout 60;
retry 60;
reboot 10;
backoff-cutoff 40;

interface "ethernet" {
send dhcp-lease-time 5000;
send dhcp-client-identifier "Galapagos";
}
```

E.2 help

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

E.2.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

E.2.2 Syntax

```
dhcpclient help <command|all>
```

E.2.3 Description

This command provides help on the various console commands provided by the ATMOS DHCP client. Specifying the command name gives detailed help, and specifying the argument *all* gives detailed help on all commands.

E.2.4 Example

```
prompt> dhcpclient help

Help is available on the following commands:

config  help                    pool            status
trace   untrace
```

E.3 pool

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

E.3.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

E.3.2 Syntax

```
dhcpclient pool [verbose]
```

E.3.3 Description

This command displays the state of the memory pool being used by the DHCP client. Should the client ever run out of memory, use of this command is helpful in determining the optimum memory pool size for the client. For example, supporting DHCP client functionality on several interfaces simultaneously will require proportionately more memory. The default pool size specified in the system file *dhcpclient* is 40000 bytes.

The verbose option lists all allocated and freed memory chunks.

E.3.4 Example

```
prompt> dhcpclient pool
DHCP Client Memory Pool Status
total pool size  39968
free             21392
allocated        18576
mean alloc chunk 67
max free chunk   13904
```

E.4 status

Note - This console command **has** been replaced by the CLI command *dhcpclient show* on page 98. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

E.4.1 Syntax

```
dhcpclient status [all]
```

E.4.2 Description

This command provides DHCP status information for the active bound lease associated with each valid interface in turn, including IP address, time until lease renewal, subnet mask and DHCP server address. Including the *all* option shows, for each valid interface, the active lease, leases which are being, or have been offered to the interface, and any leases which are still being held by the client which are not currently active (since a single interface can only have one active lease at a time).

E.4.3 Example

```
prompt> dhcpclient status
DHCP Client Lease Status (active lease only)
Interface 'ethernet'
Status    | Server ID       | IP address      | Subnet mask     | Renewal
----------+-----------------+-----------------+-----------------+-----------
*ACTIVE* | 192.168.219.151 |   192.168.219.1 |   255.255.255.0 | 31 seconds
-------------------------------------------------------------------------------
```

E.5 trace

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

E.5.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

E.5.2 Syntax

```
dhcpclient trace <trace option>
```

E.5.3 Description

This command enables or disables tracing for the DHCP client. If no arguments are given the command lists the current tracing options enabled.

The following trace options are available:

Option	Description
lease	Report changes in lease status (any interface)
bootp	Report changes in lease status (any interface)
error	Report all errors (fatal events)
warn	Report “warn” level events (important events)
note	Report “note” level events (minor/frequent events)
all	All trace options

Tracing options are disabled by using the *untrace* command with the option names to be disabled.

Saving configuration does not preserve the current tracing options that are enabled. By default tracing of *error*, *warn* and *note* are enabled.

E.5.4 Example

```
prompt> dhcpclient trace

No tracing options currently enabled.

prompt> dhcpclient trace error warn note

Currently tracing: errorwarn note
```

For more information, see *DO-007285-PS, ATMOS TCP/IP Functional Specification*.

F:DHCP-relay Console Commands

This chapter describes the DHCP-relay Console commands.

F.1 dhcprelay add

Note - This console command **has** been replaced by the CLI command *dhcprelay add* on page 792. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

F.1.1 Syntax

```
dhcprelay add [ip address]
```

F.1.2 Description

This command adds the entered IP address to the relay's list of known DHCP servers. Changes made will not come into effect until system restart. Ensure that you save configuration (using *flashfs update*) prior to restarting. A maximum of 10 DHCP server addresses can be stored by the relay.

F.1.3 Example

```
bd3000> dhcprelay add 192.168.219.7

dhcprelay: Change will have no effect until 'flashfs
update' and reboot.

bd3000>
```

F.2 dhcprelay config

Note - This console command **has** been replaced by the CLI command *dhcprelay list servers* on page 107. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

F.2.1 Syntax

```
dhcprelay config
```

F.2.2 Description

This command displays the current configuration of the DHCP relay, which comprises a list of IP addresses of known DHCP servers.

F.2.3 Example

```
bd3000> dhcprelay config

DHCP Relay - Registered DHCP Servers

-----------------------------------

192.168.219.6

bd3000>
```

F.3 dhcprelay delete

Note - This console command **has** been replaced by the CLI command *dhcprelay delete server* on page 105. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

F.3.1 Syntax

```
dhcprelay delete [ip address]
```

F.3.2 Description

This command deletes the specified IP address from the relay's list of known DHCP servers, if the named server exists. If the address is omitted, then the last server address entry in the relay's list is deleted. Changes made will not come into effect until system restart. Ensure that you save configuration (using *flashfs update*) prior to restarting.

F.3.3 Example

```
bd3000> dhcprelay delete 192.168.219.7

dhcprelay: Change will have no effect until 'flashfs
update' and reboot.

bd3000>
```

F.4 dhcprelay help

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

F.4.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

F.4.2 Syntax

```
dhcprelay help <command|all>
```

F.4.3 Description

This command provides help on the various console commands provided by the ATMOS DHCP relay. Specifying the command name gives detailed help, and specifying the argument "all" gives detailed help on all commands.

F.4.4 Example

```
bd3000> dhcprelay help
Help is available on the following commands:
config  help    pool    status  trace   untrace
```

F.5 dhcprelay pool

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

F.5.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

F.5.2 Syntax

```
dhcprelay pool [verbose]
```

F.5.3 Description

This command displays the state of the memory pool being used by the DHCP relay.

The verbose option lists all allocated and freed memory chunks.

F.5.4 Example

```
bd3000> dhcprelay pool
DHCP Relay Memory Pool Status
total pool size  10000
free             9838
allocated        162
mean alloc chunk 162
max free chunk   9838
```

F.6 dhcprelay status

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

F.6.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

F.6.2 Syntax

```
dhcprelay status
```

F.6.3 Description

This command lists the interfaces upon which the DHCP relay entity is currently listening (if the relay has at least one valid DHCP server address in its list).

F.6.4 Example

```
bd3000> dhcprelay status
DHCP Relay listening on:
ethernet
```

F.7 dhcprelay trace/untrace

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

F.7.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

F.7.2 Syntax

```
dhcprelay <trace|untrace> [trace options]
```

F.7.3 Description

This command enables or disables tracing for the DHCP relay. If no arguments are given the command lists the current tracing options enabled.

The following trace options are available:

`lease`	Report changes in lease status (any interface)
`bootp`	Report any bootp interoperation
`error`	Report all errors (fatal events)
`warn`	Report "warn" level events (important events)
`note`	Report "note" level events (minor/frequent events)
`all`	All trace options

Tracing options are disabled by using the *untrace* command with the option names to be disabled.

Saving configuration does not preserve the current tracing options that are enabled. By default tracing of *error* is enabled.

F.7.4 Example

```
bd3000> dhcprelay trace

No tracing options currently enabled.

bd3000> dhcprelay trace error warn note
```

```
Currently tracing: errorwarnnote
```

F.8 dhcprelay version

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

F.8.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

F.8.2 Syntax

```
dhcprelay version
```

F.8.3 Description

The version command displays the current ATMOS DHCP software version.

F.8.4 Example

```
bd3000> dhcprelay version
ATMOS DHCP Version 1.07
bd3000>
```

G:DHCP-server Console Commands

This chapter describes the DHCP-server Console commands.

G.1 config

Note - This console command **has** been replaced by the CLI command *dhcpserver show* on page 133. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

G.1.1 Syntax

```
dhcpserver config [add <text>|confirm|delete|flush]
```

G.1.2 Description

This command displays or edits the current configuration of the DHCP server. To display current configuration, provide no arguments to the command.

- Use of the *add* option adds the line <*text*> to the configuration file.
- Use of the *confirm* option re-parses the configuration file, confirming the changes made if the parse is successful.
- Use of the *delete* option deletes the last line from the configuration file.
- Use of the *flush* argument deletes the whole configuration.

Following any change to the configuration file, it is necessary to **confirm** the changes, issue a *flashfs update* command to commit the change to FLASH, and then restart the system before the changes can take effect.

G.1.3 Example

```
prompt> dhcpserver config
---
Current DHCP server configuration
---
allow unknown-clients;
allow bootp;


subnet 192.168.219.0 netmask 255.255.255.0 {
range 192.168.219.10 192.168.219.30;
```

```
max-lease-time 5000;
}
prompt> dhcpserver config flush
Configuration file flushed.
prompt> dhcpserver config
---
Current DHCP server configuration
(Issue "dhcpserver config confirm" followed by "flashfs
update" to confirm new configuration)
---
prompt>
```

G.2 help

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

G.2.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

G.2.2 Syntax

```
dhcpserver help <command|all>
```

G.2.3 Description

This command provides help on the various console commands provided by the ATMOS DHCP server. Specifying a command name gives detailed help on the command. Specifying *all* gives detailed help on all available commands.

G.2.4 Example

```
prompt> dhcpserver help

Help is available on the following commands:

config  help                    pool            status
trace   untrace
```

G.3 pool

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

G.3.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

G.3.2 Syntax

```
dhcpserver pool [verbose]
```

G.3.3 Description

This command gives a summary of DHCP server memory usage. The verbose option shows the entire memory allocation/free list.

G.3.4 Example

```
prompt> dhcpserver pool
DHCP Server Memory Pool Status
total pool size                          79968
free                                     52448
allocated                                27520
mean alloc chunk                         59
max free chunk                           30416
```

G.4 reset

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

G.4.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

G.4.2 Syntax

```
dhcpserver reset
```

G.4.3 Description

This command prompts the server to do a *warm* reset of itself. This has the effect of bringing the server back up **as if** the system had been rebooted, except that the lease database is preserved in SDRAM between resets.

Please note, however, you should still save the configuration file to FLASH if you want the configuration to be preserved upon rebooting the whole system.

The advantage of this command is that it allows configuration changes that have been confirmed (using *config confirm*) to take effect immediately, rather than having to do a *flashfs update* and *restart*.

This command is also convenient for defining subnet topologies for IP interfaces that have been added dynamically.

G.4.4 Example

```
prompt> dhcpserver reset

dhcpserver: Reset request acknowledged. Reset imminent.
```

G.4.5 See also

config on page 802.

G.5 status

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

G.5.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

G.5.2 Description

This command provides a summary of all leases known to the server on each interface in turn. It also shows remaining available IP addresses (i.e. those with no specified lease time, or client identifier).

G.5.3 Example

```
prompt> dhcpserver status
DHCP Server Lease Status


Interface "ethernet"
IP address      | Client UID/hw addr     | Expiry
----------------+------------------------+-----------------
  192.168.219.1 | 01:00:20:af:20:6f:59   | 11 hours
  192.168.219.2 | 01:00:20:af:11:2a:ac   | 8 hours
  192.168.219.3 |               Myclient | 140 seconds
  192.168.219.4 | 00:20:af:20:00:2b      | 2 days
  192.168.219.5 |              <unknown> | Never
  192.168.219.6 |              <unknown> | Never
  192.168.219.7 |              <unknown> | Never
  192.168.219.8 |              <unknown> | Expired
  192.168.219.9 |              <unknown> | Expired
 192.168.219.10 |          Foobarbozzle | Expired
```

G.6 trace

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

G.6.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

G.6.2 Syntax

```
dhcpserver trace <trace option>
```

G.6.3 Description

This command enables or disables tracing for the DHCP server. If no arguments are given, the command lists the current tracing options enabled.

The following trace options are available:

Option	Description
lease	Report changes in lease status (any device)
bootp	Report any BOOTP interoperation/emulation
error	Report all errors (fatal events)
warn	Report all warnings
note	Report “note” level events (minor events)
all	All trace options

Tracing options are disabled by using the *untrace* command in the same way.

Saving configuration does not preserve the current tracing options that are enabled. By default, only tracing of *error* is enabled.

G.6.4 Example

```
prompt> dhcpserver trace
No tracing options currently enabled.
prompt> dhcpserver trace error warn note
Currently tracing: error warn note
```

G.7 version

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

G.7.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

G.7.2 Description

This command displays the current version number of the ATMOS DHCP software.

G.7.3 Example

```
prompt> dhcpserver version

ATMOS DHCP Version 1.07

prompt>
```

H:DNS Client Console Commands

This chapter describes the DNS Client Console commands.

H.1 DNS Client Console Commands

The DNS client allows some configuration parameters to be set from the console. This includes the name server IP address list and the search list. The name server IP address list is a list of the IP addresses for the default name servers used by the DNS client. The search list is a list of domains for the DNS client to try for incomplete host names. This allows the use of host names like *host* when the complete name would be *host.school.edu*.

H.2 nameserver

Note - This console command **has** been replaced by the CLI commands *dnsclient add server* on page 150 and *dnsclient delete server* on page 154. Use the CLI commands instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use CLI commands.

H.2.1 Syntax

```
nameserver [ipaddress]|delete <ipaddress>
```

H.2.2 Description

This command adds or deletes a nameserver IP address to the default nameserver list. The address should be in dotted decimal form. Note the DNS client cannot operate normally without at least one default nameserver address.

H.2.3 Example

```
192.168.219.196 dnsclient> nameserver 10.45.20.1
192.168.219.196 dnsclient> nameserver 10.45.25.2
192.168.219.196 dnsclient> nameserver 10.45.25.1
192.168.219.196 dnsclient> nameserver delete 10.45.25.1
192.168.219.196 dnsclient>
```

H.3 search

Note - This console command **has** been replaced by the CLI command *dnsclient add searchdomain* on page 149. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

H.3.1 Syntax

```
search <domain_1 domain_2 . . . domain_n>
```

H.3.2 Description

This command will replace the current, or create a, domain search list. The DNS client uses this list when a user asks for the IP address list for an incomplete domain name. For example if the search list consists of *cs.school.edu* and *school.edu* and the user requests the IP address list for host name *dns* the DNS client would try *dns.cs.school.edu* and if that failed would then try *dns.school.edu*. The list is not used when the user specifies a complete domain name such as *www.globespanvirata.com*. The search list is a white space separated list of domain names and allows for one to N entries.

H.3.3 Example

```
192.168.219.196 dnsclient> search cs.school.edu school.edu
192.168.219.196 dnsclient>
```

H.4 show

Note - This console command **has** been replaced by the CLI command *dnsclient list searchdomains* on page 155 and *dnsclient list servers* on page 156. Use the CLI commands instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use CLI commands.

H.4.1 Syntax

```
show
```

H.4.2 Description

This command will display the current domain configuration data used by the DNS client.

H.4.3 Example

```
192.168.219.196 dnsclient> show
search cs.school.edu school.edu
nameserver 10.45.20.1
nameserver 10.45.25.1
192.168.219.196 dnsclient>
```

H.5 nslookup

Note - This console command **has** been replaced by the CLI commands *dnsclient add searchdomain* on page 149 and *dnsclient add server* on page 150. Use the CLI commands instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use CLI commands.

H.5.1 Syntax

```
nslookup lookup_argument
```

H.5.2 Description

This command will do a DNS lookup on the argument. If the argument is a host name an IP address will be displayed, or if the argument is an IP address a host name will be displayed. Note that either form of the command can fail. If the command does fail an appropriate error message will be displayed. The DNS client's cache is searched before any name server query is sent on the network.

H.5.3 Example

```
192.168.219.196 dnsclient> nslookup yahoo.com
Name:  yahoo.com
Address[0]: 216.115.108.243
Address[1]: 216.115.108.245
192.168.219.196 dnsclient>
```

H.6 cache

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

H.6.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

H.6.2 Syntax

```
cache [flush [all]|[argument]][show]
```

H.6.3 Description

This command allows displaying the entire cache or deleting the entire cache or specific entries. An entry can be removed by issuing the *cache flush* command with an IP address or a host name. The *yahoo.com* entry could be removed by using *cache flush 216.115.108.245* or *cache flush yahoo.com*, note that the name must match exactly.

H.6.4 Example

```
192.168.219.196 dnsclient> cache show
Hostname yahoo.com.
Entry has 86362 seconds to live.
Number addresses = 2.
Address[0] = 216.115.108.243
Address[1] = 216.115.108.245

Hostname ns0.globespanvirata.com.
Entry has 86362 seconds to live.
This entry is for a Name Server.
Number addresses = 1.
Address[0] = 192.168.219.9
```

```
192.168.219.196 dnsclient> cache flush all
192.168.219.196 dnsclient>
```

I: DNS Relay Console Commands

This chapter describes the DNS Relay Console commands.

I.1 dnsrelay config

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

I.1.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

I.1.2 Syntax

```
dnsrelay config [reset]
```

I.1.3 Description

This command displays the configuration of the DNS relay, including the DNS server address, the number of communication retries the relay will attempt in the event of a failed connection, and whether or not the relay has managed to connect results in the configuration being reset to factory default settings.

I.1.4 Example

```
prompt> dnsrelay config

Server discovery mode: MANUAL

DNS Server address: 192.168.96.200 - Connected

Max connection retries: 3

prompt> dnsrelay config reset

dnsrelay: Default settings restored. (Warning: Must re-
connect to DNS server,

dnsrelay: all old outstanding traffic and connections will
be dropped).
```

I.2 dnsrelay help

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

I.2.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

I.2.2 Syntax

```
dnsrelay help [command|all]
```

I.2.3 Description

This command provides help on the various console commands provided by the ATMOS DNS relay. Specifying the command name gives detailed help, and specifying the argument *all* gives detailed help on all commands.

I.2.4 Example

```
prompt> dnsrelay help

Valid DNS relay commands are:

   config       help       pool      retry     server
   status      trace    untrace    version
```

I.3 dnsrelay pool

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

I.3.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

I.3.2 Syntax

```
dnsrelay pool [verbose]
```

I.3.3 Description

This command displays the state of the memory pool being used by the DNS relay.

The *verbose* option lists all allocated and freed memory chunks.

I.3.4 Example

```
prompt> dnsrelay pool
DNS Relay Memory Pool Status
total pool size9968
free9872
allocated96
mean alloc chunk32
max free chunk9856
```

I.4 dnsrelay retry

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

I.4.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

I.4.2 Syntax

```
dnsrelay retry <retry value>
```

I.4.3 Description

This command sets the maximum number of retries the DNS relay is allowed to perform in the event of connection or transmission failure. the retry value must be a number between 1 and 10.

I.4.4 Example

```
prompt> dnsrelay retry 4
Connection retry value set to 4.
prompt>
```

I.5 dnsrelay server

Note - This console command **has** been replaced by the CLI command *dnsrelay list servers* on page 162. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

I.5.1 Syntax

```
dnsrelay server <DNS server IP address>
```

I.5.2 Description

This command tells the DNS relay which DNS server to contact. Caution must be exercised when using this command - if the DNS relay already knows which DNS server to contact then all existing connections will be reset, all outstanding traffic dropped, and the relay will then attempt to communicate with the newly appointed DNS server.

I.5.3 Example

```
prompt> dnsrelay server 192.168.219.50
DNS server address set to 192.168.219.50.
prompt>
```

I.6 dnsrelay status

Note - This console command **has** been replaced by the CLI command *dnsrelay list servers* on page 162. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

I.6.1 Syntax

```
dnsrelay status
```

I.6.2 Description

This command displays the status of the DNS relay, including whether or not it knows which DNS server to try and contact and, if so, whether or not it has successfully connected to the server.

I.6.3 Example

```
prompt> dnsrelay status
DNS relay status
DNS server address discovery incomplete.
prompt> dnsrelay server 192.168.219.50
DNS server address set to 192.168.219.50.
prompt> dnsrelay status
DNS relay status
DNS server address: 192.168.219.50
Connection status: Connected
prompt>
```

I.7 dnsrelay trace/untrace

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

I.7.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

I.7.2 Syntax

```
dnsrelay <trace|untrace> [trace options]
```

I.7.3 Description

This command enables or disables tracing for the DNS relay. If no arguments are given, the command lists the current tracing options enabled.

The following trace options are available:

```
socketReport ALL socket-related I/O

queryTrace DNS resolver queries

responseTrace DNS server responses

errorReport all serious, error-level events

warnReport all minor, warning-level events

connTrace DNS server connectivity

allActivate all trace options
```

Trace options are disabled by using the *untrace* command with the option names to be disabled.

Saving configuration does not preserve the current tracing options that are enabled. By default tracing of *error* is enabled.

I.7.4 Example

```
prompt> dnsrelay trace

No tracing options currently enabled.
```

```
prompt> dnsrelay trace error warn query
Currently tracing: error warn query
```

I.8 dnsrelay version

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

I.8.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

I.8.2 Syntax

```
dnsrelay version
```

I.8.3 Description

The *version* command displays the current ATMOS DNS relay software version.

I.8.4 Example

```
prompt> dnsrelay version
ATMOS DNS Relay Version 1.01
prompt>
```

J: TCP/IP Console commands

This chapter describes the TCP/IP Console commands.

J.1 Console commands obsoleted in ISOS 8.1

The following table lists the console commands that were supported in previous releases, but are not supported by the IP stack in releases ISOS 8.1 and later.

Obsolete console command	
ipatm pvc	ip restart
ipatm lifetime	ip rip accept
ipatm help	ip rip allowed
ipatm flies	ip rip boot
ipatm arpserver	ip rip help
ipatm arp	ip rip hostroutes
ipatm abort	ip rip killrelay
ip abort	ip rip poison
ip arp	ip rip relay
ip arprouting	ip rip relays
ip autoloop	ip rip rxstatus
ip disable	ip rip send
ip error	ip rip trigger
ip etherfiles	ip route
ip files	ip routeflush
ip flush	ip routes
ip get	ip stats
iphostname	ip subnet
nat	ip trace
ip noerrors	ip untrace
ip norelay	ip uptime
ip ping	?
ip protocols	ip relay

J.2 config

Note - This console command **has** been replaced by the CLI commands *ip show* on page 671 and *ip list interfaces* on page 648. Use the CLI commands instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use these CLI commands.

J.2.1 Syntax

```
config [save]
```

J.2.2 Description

Displays the IP configuration (not including the *snmp* configuration), or saves it in flash memory.

The functionality of the *config* command is also accessible in the standard way through the config process (eg. *config print ip*), if that process is present. However, when accessed through the config process, the *snmp* configuration *is* included.

J.2.3 Example

```
prompt> ip config

192.168.88.50> ip config

device add ipoa       atmpvc    //atm     mtu 1500

device add ethernet   ether  //bun/port=etherfilter mtu
1500  192.168.88.50

subnet add ethernet.home  . 192.168.88.0     ff:ff:ff:00

rip send    ipoa     2

rip send    ethernet none

rip accept  ipoa     1 2

rip accept  ethernet none

ipatm lifetime 60

# IP host table:

# Port table:

l2tp       1701/UDP

router     520/UDP

snmp       161/UDP
```

```
tftp        69/UDP
telnet      23/TCP
prompt> ip config save
Updating flash filing system ...
done
ip: configuration saved
```

J.2.4 See also

snmp on page 842.

The other commands which are used for setting configuration displayed and saved by *config*, are listed below:

- *device* on page 833.
- *portname* on page 840.

J.3 device

Note - Some device console commands **have** been replaced by CLI commands; *device add* has been replaced by *ip add interface* on page 623, *ip set interface dhcp* on page 656, and *ip set interface mtu* on page 655, *device delete* has been replaced by *ip delete interface* on page 637 and *device list* has been replaced by *ip list interfaces* on page 648. Use the CLI commands instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use these CLI commands.

Note - The console command *device flush* **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

J.3.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

J.3.2 Syntax

```
device [list]

device add <i/f> <type> [<file>] [mtu <size>] [<IP
address>]

device delete <i/f>

device flush
```

J.3.3 Description

Displays the interfaces that IP is configured to use (*device list*), or adds an interface to the configuration (*device add*), or deletes an interface (*device delete*), or deletes all interfaces (*device flush*), from the configuration.

The options used with this command are described below:

- <*i/f*> is an arbitrary label for the interface, which is used in referring to it in subsequent commands. (It is often chosen to be the same as <*type*>, though this is perhaps slightly confusing.)

- <*type*> specifies the class of interface: Ethernet-*like*, IP-over-ATM, PPTP or loopback.

  The supported values for <*type*> are shown in the table below:

Class	<Type>	Default file
Ethernet	ether	*value of* ETHERNET_DEVICE_NAME (defined in system file)
Ethernet	ether	*//edd*
	flane	*//lec1*
	bridge	*//bridge*
IP-over-ATM	atmpvc	*//bun* *//atm*
	atm	*//q93b*
Point-to-point	PTP	*None*
	PPP	*//ppp/DEVICE=1*
Loopback	loop	-

  A default file for the *Ethernet* class can be defined in the system file, as follows:

  `config.h ETHERNET_DEVICE_NAME "s//edd"`

  If a default file is not defined, the name *ether* is not supported. However, it is still possible to define devices of type *ether* with an explicit filename.

  The class *IP-over-ATM* includes both SVC-based and PVC-based IP-over-ATM; the decision whether to use SVCs or PVCs is made at initialization, by testing the interface colors of the file if it supports the *Indigo* interface, then SVCs are used, and otherwise PVCs.

- <*file*> specifies the file name that will be opened to access the underlying device. The device can be any of the following:
  - Ethernet
  - IP-over-ATM
  - PTP
  - Loopback

The device **must** provide the colored interface appropriate for that type of device.

For a loopback interface, <*file*> is not used, and can just be specified as "-" or omitted altogether.

Note that several different values of <*type*> specify the same class of interface; they differ in that each implies a different default value for <*file*>. As a result, for the most common interface configurations, <*file*> can be omitted, and one need only specify the appropriate value of <*type*>.

- <*mtu*> specifies the MTU (maximum transmission unit); that is, the size of the largest datagram (excluding media-specific headers) that IP will attempt to send through the interface. The value specified will be ignored if it is larger than the maximum supported by the interface class, which is currently 1500, unless the *IP-over-ATM MTU* value has been changed in the TCP/IP build-time configuration system file. Normally, there is no point in setting the MTU less than this, so the <*mtu*> option is of little use.
- <*IP address*> is the IP address that this system uses on the interface; if it is not specified, the interface will be disabled until an IP address is supplied with the *ip enable* command.

  For a loopback interface, the address should be set to *127.0.0.1*. (All addresses of the form 127.*.*.* will then be recognized as loopback addresses, as is normal practice.)

  For non-loopback interfaces, the subnet mask for the local network will be assumed to be *ff:ff:ff:00* (e.g. a class C network); if the correct subnet mask is other than this then it will need to be set with the *subnet* command.

  If there is a DHCP client in the system, the address can be set to DHCP. This setting means that the IP address should be *learned* by DHCP. For example:

  ```
  ip device add ether ether dhcp
  ```

  Note that DHCP is not supported on all interface types.

If the IP process is given a command line (a little-used feature of ATMOS!) then each argument will be treated as a possible Ethernet-like file to open, given names *ether1*, *ether2*, and so on.

For example, if the IP process is defined in the system file as:

```
Process ip is tcp_ip/ip //bridge //lec1
```

(and *//bridge* and *//lec1* can be opened), then the equivalents of the commands:

```
device add ether1 ether //bridge
device add ether2 ether //lec1
```

will be processed, in addition to the others above.

Configuration saving saves the interface configuration.

J.3.4 Example

```
prompt> ip device
#                 type   dev file        IP address
device ether      ether  //edd mtu 1500  192.168.2.1
device ipoa      atmpvc //atm mtu 1500   192.168.55.1
prompt> ip device add loop loop 127.0.0.1
prompt> ip device delete ipoa
prompt> ip device
#                 type   dev file         IP address
device ether      ether  //edd   mtu 1500 192.168.2.1
device ipoa      atmpvc  //atm   mtu 1500 192.168.55.1
device loop       loop   -           mtu 2048  127.0.0.1
```

J.3.5 See also

enable on page 837

J.4 enable

Note - This console command **has** been replaced by CLI commands *ip attach* on page 632, *ip set interface mtu* on page 655 and *ip set interface ipaddress* on page 652. Use the CLI commands instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use CLI commands.

J.4.1 Syntax

```
enable [<i/f> [mtu <size>] [<IP address>]]
```

J.4.2 Description

Enables all interfaces, or just a specified interface. Can also be used to set the MTU and IP address on an interface when enabling it (or change them on an interface that is already enabled); see *device* on page 833 for details on the interfaces.

Configuration saving saves the MTU and IP addresses, but not the disabled/enabled state.

J.4.3 Example

```
prompt> ip enable etherent 192.168.2.1
ip/ethernet: IP address 192.168.56.3
prompt> ip device
#                 type    dev file        IP address
device ether      ether   //edd mtu 1500   192.168.2.1
```

J.4.4 See also

device on page 833

J.5 help

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

J.5.1 CLI access permission - usable commands

Users with the following access permissions can use this command:

- Engineer
- Super user

J.5.2 Syntax

```
help
help <cmd>
help all
```

J.5.3 Description

Displays a summary of available commands, more detailed information on a particular command, or more detailed information on all commands.

(Some commands are *hidden* and are not displayed by *help* or *help all*; help is still available on these using the *help* <*cmd*> form if you know the name of the command.)

J.5.4 Example

```
prompt> ip help
Commands are:
?       arp     config   device
disable enable  help     ipatm
nat     norelay ping     relay
rip     route   routes   snmp
stats   subnet  version
'.' repeats the last command
Type "ip help all" or "ip help <command>" for more details
```

```
prompt> ip help arp
arp Syntax
  arp <cmd>         - execute arp subcommand
  arp help          - list subcommands available
```

J.6 portname

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

J.6.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

J.6.2 Syntax

```
portname add <name> <number>[/<protocol>]

portname flush

portname list

portname read <file>

portname help [all|<cmd>]
```

J.6.3 Description

Sets up a mapping between a UDP or TCP port and a symbolic name; deletes all such mappings; lists the mappings; reads the mappings from a file; or displays help on the “portname” command.

The symbolic names can be used as values of the attributes LPORT and RPORT (described in the *ATMOS TCP/IP Functional Specification, DO-007285-PS*) provided the protocol type (UDP or TCP) is appropriate. They are also displayed in place of port numbers, when a suitable mapping exists. The Damson interface (described in the *ATMOS TCP/IP Functional Specification, DO-007285-PS*) allows other processes to query the mapping.

<protocol> should be either *UDP* or *TCP*; it can be omitted, but that is not very useful.

For *portname read*, the file is in the same format as //isfs/services (described in the *ATMOS TCP/IP Functional Specification, DO-007285-PS*), which is the same as the output from *portname list*.

The *portname* command is *hidden*, not shown by *ip help*.

Configuration saving saves this information.

J.6.4 Example

```
prompt> ip portname flush
prompt> ip portname add someport 105/tcp
prompt> ip portname list
someport   105/TCP
prompt> ip portname read //isfs/services
prompt> ip portname list
router     520/UDP
snmp       161/UDP
tftp       69/UDP
telnet     23/TCP
someport   105/TCP
```

J.7 snmp

Note - The console commands *ip snmp access list, ip snmp config* and *ip snmp trap list* **have not** been replaced by CLI commands. If used, these commands will not create inconsistencies between the Information Model and the underlying system. They are *usable* console commands.

J.7.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

Note - The console commands *snmp access read/ write/ delete/ flush, snmp help, snmp trap add/ delete/ flush, snmp config save and snmp version* **have not** been replaced by CLI commands. If used, these commands will create inconsistencies between the Information Model and the underlying system. They are *blacklisted* console commands that can only be used **with extreme caution**.

J.7.2 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

J.7.3 Syntax

```
snmp access [read|write|delete|flush|list] <parameters>

snmp config [save]

snmp help [<cmd>|all]

snmp trap [add|delete|flush|list] <parameters>

snmp version
```

J.7.4 Description

Manages the list of SNMP community names (also used as passwords by other applications, such as *telnet*) and the list of SNMP trap destinations. (See the *ATMOS TCP/IP Functional Specification, DO-007285-PS* for information about the interface to this information.)

The syntax of the commands is documented in the *ATMOS SNMP Functional Specification, DO-007285-PS*.

The *snmp version* command displays the version number of ATMOS SNMP. Note, the version number returned is the internal version number of GlobespanVirata code, **not** the version of the SNMP protocol supported, which is SNMP v1.

Note – In standard ATMOS systems the console is configured to allow the commands to be accessed by typing just *snmp ...* instead of *ip snmp ...* at the command line.

J.8 version

Note - This console command **has** been replaced by CLI command *version* on page 50. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

J.8.1 Syntax

```
version
```

J.8.2 Description

Displays the version of the IP process, ATM address, and MAC address.

(An obsolescent option *ip* still exists, but *version ip* now displays misleading information and should not be used.)

J.8.3 Example

```
prompt> ip version

IP version 1.54

ATM address:
47.00.83.10.a2.b2.c2.00.00.00.00.00.00.00.20.2b.00.00.38.0
0

MAC address:   0:20:2b:0:0:38
```

K:PPP Console Commands

This chapter describes the PPP Console commands.

K.1 Console object types

The **ppp** process presents its setup in terms of a number of distinct object types:

- The upper limit on the number of each of these objects permitted in a system is configured using the *config resource* console command.
- The current state of each object is saved by *config save*.

K.1.1 Channels

The **ppp** process provides a number of PPP connection *channels*. A channel is a single PPP connection. Channels are numbered from 1. Many **ppp** console commands affect only a single channel. The command is prefixed with the channel number.

K.1.2 Users

A *user* is a user name and password. All users must have distinct names. The *user* console command controls these.

K.1.3 Tunnels

A *tunnel* is a PPTP or L2TP connection. Tunnels are numbered from 1. PPP channels must be associated with a tunnel to be involved in PPP tunnelling. The *tunnel* console command provides control of these.

K.1.4 Interfaces

An interface is an internal MAC (Ethernet) device. PPP channels must be associated with an interface to be involved with bridging or routing.

K.1.5 Interface 1 and Channel 1

Interface 1 has some special functions associated with it, allowing dynamic IP address assignment to be performed. Channel 1 is by default associated with Interface 1. These two should be used only for IP dial-out functions, and for this function should be attached to the router interface named *ppp_device*. The dial-out example in the following section makes this clearer. These specializations have been made to make the configuration of an IP dial-out simpler.

K.2 Console examples

K.2.1 Simple test

The simplest thing you can do to test ATMOS PPP, between two PPP channels in a single ATMOS system, is to create a PVC in the switch to which the test box is connected, between two VCIs (say 32 and 33 here) on the connected switch port. Type the following:

```
pvccreate a1 32 a1 33
```
(at the switch console, if it is a Virata Switch)

```
ppp event 5
```
(at the console of the PPP ATMOS system)

```
ppp 1 pvc 32
ppp 2 pvc 33

ppp 1 enable
ppp 2 enable
```
(they should now swap packets and synchronise)

```
ppp 1 status
```

This should show that the two ends are connected. No data will be exchanged.

K.2.2 IP dial-out over PPP

To perform a dial-out over a PVC, operate as follows:

First set up a router device for PPP to use. No IP address should be specified, so that the device is created but not enabled. The device name *ppp_device* should be used.

```
ip device add ppp_device ether //ppp/DEVICE=1

ppp 1 pvc <whatever>
ppp 1 welogin <name> <password>
ppp 1 enable
```

If the configuration is saved at this point, the dial-in will be attempted automatically.

K.2.3 IP dial-in server setup

To create a system which can accept dial-in connections over PVCs, type the following:

Note – For a complex setup such as this, it may be more convenient to create it on another system using a text editor, then TFTP the setup into the ATMOS system.

Note – Assume that 8 dial-in PVCs are being created, numbered 32 to 39. These will be created as channels 2 to 9. A single IP subnet will be created, attached to a port of the router via interface 3. The IP subnet 192.168.200.0 will be used, with channel n assigning address 192.168.200.n to the far end. The server interface will take address 192.168.200.99.

```
ip device add ppp_device3 ether //ppp/DEVICE=3
192.168.220.99
```

Further IP setup may be needed, for instance to route the result to some other device such as the Ethernet port.

```
ppp interface 3 localip 192.168.220.99

ppp 2 theylogin chap

ppp 2 pvc 32 listen

ppp 2 interface 3

ppp 2 remoteip 192.168.200.2

ppp 2 enable
```

(and the corresponding setup for each of the channels 3 to 8 as well)

Clients can now dial in to this server, be allocated IP addresses and traffic will be sent to and from the router.

K.2.4 Remote Bridging

To create a system where two bridges are connected by a PVC, do the following at each end: (In this example, interface 2 is attached to the bridge in ATMOS (interface 1 is reserved for routed traffic).)

```
bridge device add ppp/DEVICE=2
```

(Attach interface 2 to the bridge.)

```
ppp 1 pvc 32 mac
```

```
ppp 1 interface 2
ppp 1 enable
```

If required, multiple interfaces can be attached to the bridge of a single ATMOS system so that a single *master* site is bridged to several satellites. Each incoming bridging PPP channel should be attached to a distinct interface. Each interface must be independently attached to the bridge.

K.3 <channel> clear

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

K.3.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

K.3.2 Syntax

```
<channel> clear
```

K.3.3 Description

Clear all aspects of this channel back to their default settings. If there is an active connection, it is torn down.

K.4 <channel> disable

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

K.4.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

K.4.2 Syntax

```
<channel> disable
```

K.4.3 Description

Clear the enable flag for a PPP channel. This is the default setting. Disabling does not remove other configured information about this channel.

In the PPP state machine, this sets the PPP link to *closed*. If it is already closed, there is no effect.

Configuration saving saves this information. By default, all channels are disabled.

K.4.4 See also

```
<channel> enable
```

K.5 <channel> discard

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

K.5.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

K.5.2 Syntax

```
<channel> discard [<size>]
```

K.5.3 Description

Discard is a PPP LCP packet type, which is like the Echo packet type but does not generate a return. This can be used for more careful tests of data transfer on the link, for instance at sizes near the negotiated MRU.

This command sends an LCP Discard packet, of the specified size. If no size is given, a minimal sized packet is sent.

Arrival of a *Discard packet* is logged locally as a level 2 event.

The link must be up and operational in order to do the discard test.

K.5.4 See also

```
<channel> echo
```

K.6 <channel> echo

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

K.6.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

K.6.2 Syntax

```
<channel> echo [<size>]
```

K.6.3 Description

Echo is an LCP packet, which is used to test an established PPP link. It solicits a ping-like reply from the far end.

This command sends an LCP Echo packet, of the specified size. If no size is given, a minimal sized packet is sent. If a size greater than the remote Maximum Receive Unit size is specified, the value is reduced to the remote MRU before sending.

The command waits for 1 second for a reply packet to arrive, and prints whether the reply arrived. If a reply arrives subsequent to this, it is logged as a level 2 event.

The link must be up and operational in order to do the echo test.

K.6.4 See also

<channel> discard on page 852.

K.7 <channel> echo every

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

K.7.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

K.7.2 Syntax

```
<channel> echo every <seconds>
```

K.7.3 Description

Echo is an LCP packet, which is used to test an established PPP link. It solicits a ping-like reply from the far end.

This command sets a channel to confirm the continued presence of an open PPP connection by sending an LCP echo every few seconds, and requiring an echo reply. The number of seconds between echo requests is specified as a parameter.

If 0 is specified, the function is disabled. Use the *info all* command to read the current state on a channel.

Configuration saving saves this information. By default, the function is disabled.

K.7.4 See also

<channel> echo on page 853 (manually initiated LCP echo) and *<channel> info* on page 858 (using the *all* option to show current state).

K.8 <channel> enable

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

K.8.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

K.8.2 Syntax

```
<channel> enable
```

K.8.3 Description

Set the enable flag for a PPP channel. By default, this is disabled.

In the PPP state machine, this flag sets the PPP link to *open*. If it is already open, there is no effect.

Configuration saving saves this information. By default, all channels are disabled.

K.8.4 See also

<channel> disable on page 851 (reverse the effect).

K.9 <channel> event

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

K.9.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

K.9.2 Syntax

```
<channel> event [<n>]
```

K.9.3 Description

Read or set the overall trace output level.

Configuration saving does not save this value. The default event level is 1.

The event levels are shown in the table below:

Level	Description
1	Only very serious errors reported
2	Definite protocol errors or very significant events reported.
3	Links going up/down reported.
4	Every packet and significant state change is reported.
5	Every packet sent/received is disassembled, and hex dumped.

K.10 <channel> hdlc

Note - This console command **has** been replaced by the CLI command *pppoa set transport headers hdlc* on page 344. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

K.10.1 Syntax

```
<channel> hdlc [1|0]
```

K.10.2 Description

If *1*, use an HDLC header on the front of transmitted packets and require one on received ones. This consists of two bytes, FF-03, and assists in interoperability with some other (non-standard) implementations. If *0*, disable this.

Call with no argument to find the current setting.

The default value is 0 (disabled).

Configuration saving saves this information.

If not set, and a packet is received with an HDLC header, the channel goes into a *learned HDLC* mode and sends packets with the HDLC header. Thus, interoperation with HDLC-using equipment should not normally require any configuration. Learning occurs in this direction only.

Setting *hdlc* to *0* clears this learned state. Configuration saving does not save the learned state.

K.11 <channel> info

Note - This console command **has** been replaced by the CLI command *pppoa show transport* on page 378. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

K.11.1 Syntax

```
<channel> info [all]
```

K.11.2 Description

Provide information about the current settings of this channel. This includes all configured state, and also current protocol information.

Specifying *all* prints out more information.

info and *status* are synonyms.

K.12 <channel> interface

Note - This console command **has** been replaced by the CLI command *pppoa set transport interface* on page 349. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

K.12.1 Syntax

```
<channel> interface <n>
```

K.12.2 Description

Logically associate the specified channel with the specified interface.

Interface 1 is always the router port. It should be used for any PPP channel over which IPCP communication with the local system's IP router is desired. Other interfaces can be created for bridging. A single PPP channel can only be associated with a single interface, or a single tunnel.

Use *info* to find the current setting.

Calling with n=0 removes any association. This is the default state.

Configuration saving saves this information.

K.12.3 See also

<channel> info on page 858, *<channel> tunnel <n> <tunnel protocol> <dial direction>* on page 871.

K.13 <channel> lcpmaxconfigure

Note - This console command **has** been replaced by the CLI command *pppoa set transport lcpmaxconf* on page 351. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

K.13.1 Syntax

```
<channel> lcpmaxconfigure [<n>]
```

K.13.2 Description

Set the *Max-Configure* parameter for LCP, as described in *Section 4.6 of RFC1661*. This is the maximum number of Configure Requests that will be sent without reply, before assuming that the peer is unable to respond.

Call with no argument to find the current setting.

The default value is *10*.

Configuration saving saves this information.

K.14 <channel> lcpmaxfailure

Note - This console command **has** been replaced by the CLI command *pppoa set transport lcpmaxfail* on page 352. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

K.14.1 Syntax

```
<channel> lcpmaxfailure [<n>]
```

K.14.2 Description

Set the *Max-Failure* parameter for LCP, as described in *Section 4.6 of RFC1661*. This is the maximum number of consecutive *Configure Naks* that will be sent before assuming that parameter negotiation is not converging.

Call with no argument to find the current setting.

The default value is *5*.

Configuration saving saves this information.

K.15 <channel> lcpmaxterminate

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

K.15.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

K.15.2 Syntax

```
<channel> lcpmaxterminate [<n>]
```

K.15.3 Description

Set the *Max-Terminate* parameter for LCP, as described in *Section 4.6 of RFC1661*. This is the maximum number of *Terminate Requests* that will be sent without reply, before assuming that the peer is unable to respond.

Call with no argument to find the current setting.

The default value is *2*.

Configuration saving saves this information.

K.16 <channel> llc

Note - This console command **has** been replaced by the CLI command *pppoa set transport headers llc* on page 346. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

K.16.1 Syntax

```
<channel> llc [1|0]
```

K.16.2 Description

If 1, use an LLC header on the front of the transmitted packets and require an LLC header on the received packets. This consists of four bytes, FE-FE-03-CF, and is required for *PPP Over AAL5* (RFC 2364 p4) when using LLC encapsulated PPP. If *0*, disable this.

Call with no argument to find the current setting.

The default value is 0 (disabled).

Configuration saving saves this information.

If not set, and a packet is received with an LLC header, the channel goes into a *learned LLC* mode and sends packets with the LLC header. Thus, interoperation with LLC-using equipment should not normally require any configuration. Learning occurs in this direction only. Setting *hdlc* to *0* clears this learned state. Configuration saving does not save the learned state.

K.17 <channel> pvc

Note - This console command **has** been replaced by the CLI commands *pppoa set transport dialin pvc* on page 329 and *pppoa set transport dialout pvc* on page 332. Use the CLI commands instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use these CLI commands.

K.17.1 Syntax

```
<channel> pvc [[<port>] <vpi>] <vci> [ip|mac] [listen]

<channel> pvc none
```

K.17.2 Description

Attach an ATM PVC to the given PPP channel. The port can be specified (only for a multi-port device), and the VPI (default is 0), and the VCI.

The full VPI range (0-4095) and VCI range (1-65535) is supported for this command.

If a single argument *none* is supplied, any current connection is *torn down*. This is equivalent to *svc none* on the channel.

In the PPP state machine, providing a link of this form causes the link to be *up*. Note that *enable* must also be used, to allow the link to become operational.

The *ip* or *mac* indicates which form of data is transported over the connection: one of IP data (controlled by the IPCP protocol), or MAC data (for BCP). If neither is provided, *ip* is assumed.

If the channel is not linked to an interface, and the channel is for IP data, the channel is linked to interface 1. If the channel is not linked to an interface, and the channel is for MAC data, the channel is linked to interface 2.

Providing a PVC setting unsets any SVC setting. See *<channel> svc* on page 868.

It is possible for a PVC to become *down* in the PPP state machine even though the PVC is still there, for instance due to an authentication failure. If in this state, an incoming packet will cause the PPP state machine to go *up*.

If the *listen* option is specified then this is the server end of a PVC. It will not send out PPP Configure Requests until it first receives a packet over the PVC. When a connection is torn down it goes returns to this state.

Use the *info* command to read this information.

Configuration saving saves this information. By default, a channel has no connection information.

K.17.3 Example

```
ppp 3 pvc 3 32
```
set channel 3 to be (VPI=3, VCI=32)

```
ppp 4 pvc
```
read PVC settings for channel 4

```
ppp 5 pvc 0
```
remove any PVC settings from channel 5

K.18 <channel> qos

Note - This console command **has** been replaced by the CLI commands *pppoa set transport qosclass* on page 361, *pppoa set transport pcr* on page 359 and *pppoa set transport bt* on page 327. Use the CLI commands instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use these CLI commands.

K.18.1 Syntax

```
<channel> qos [cbr|ubr] [pcr <pcr-tx> [<pcr-rx>]]
```

K.18.2 Description

Specify that the VC for a PPP channel should be Constant Bit Rate or Unspecified Bit Rate, and (optionally for UBR) give a Peak Cell Rate for the connection. If two values are specified then they are the transmit and receive PCRs respectively.

If called while not attached to a VC then the settings are saved for use when a VC is created.

If the channel is already attached to a VC then it is closed, and re-opened with the new values. If it cannot be reopened, it remains closed.

Configuration saving saves this information. By default, channels are established UBR.

K.18.3 Example

For example, to set channel 3 to be CBR limited at 10000 cells/sec, enter:

```
ppp 3 qos cbr pcr 10000
```

K.19 <channel> remoteip

Note - This console command **has** been replaced by the CLI command *pppoa set transport remoteip* on page 365. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

K.19.1 Syntax

```
<channel> remoteip [<ipaddress>]
```

K.19.2 Description

If a PPP link is established using IPCP, this call causes the channel to provide the given IP address to the remote end of the connection. PPP will refuse to complete the connection if the other end will not accept this.

This is normally used for channels on which the remote party dials in, to allocate the IP address to that remote party.

Call with no argument to find the current setting.

Call with 0.0.0.0 to remove any setting. This is the default state.

Configuration saving saves this information.

K.19.3 See also

interface <n> localip on page 874.

K.20 <channel> svc

Note - This console command **has** been replaced by the CLI commands *pppoa set transport dialin svc* on page 331 and *pppoa set transport dialout svc* on page 334. Use the CLI commands instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use these CLI commands.

K.20.1 Syntax

```
<channel> svc listen [ip|mac]

<channel> svc addr <addr> [ip|mac]

<channel> svc none
```

K.20.2 Description

Specify that the VC for a PPP channel should be an SVC (i.e. created by signalling). This can either be by listening for an incoming call, or by making an outgoing call to a specified ATM address.

The outgoing call or listen occurs immediately. If the call fails it will be retried after a few seconds. In the PPP state machine, providing a connection of this form causes the channel to be *up* or *down*. Note that *enable* must also be used, to allow the link to become operational.

Outgoing and incoming UNI signalling calls are identified by a BLLI value that identifies PPP. (Note: A BLLI of length 3 bytes is used, hex values 6B, 78. C0.)

If the channel is already attached to an SVC or PVC then it is closed, and re-opened with the new settings. If it cannot, it remains closed.

If a single argument *none* is supplied, any current connection is torn down. This is equivalent to *pvc none* on the channel.

The *ip* or *mac* option indicates which form of data is transported over the connection: one of IP data (controlled by the IPCP protocol), or MAC data (for BCP). If neither is provided, *ip* is assumed.

Providing an SVC setting unsets any PVC setting. (See the command, *<channel> pvc* on page 864.)

Configuration saving saves this information. By default a channel has no connection information.

K.20.3 Example

```
ppp 3 svc 47.00.83.01.03.00.00.00.00.00.00.00.00.00.20.2b.00.03.0b.00
ppp 4 svc listen
```

(listen for incoming call)

```
ppp 7 svc none
```

(tear down connection, remove setting)

K.20.4 See also

<channel> pvc on page 864.

K.21 <channel> theylogin

Note - This console command **has** been replaced by the CLI command *pppoa set transport theylogin* on page 371. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

K.21.1 Syntax

```
<channel> theylogin pap|chap|none
```

K.21.2 Description

This command describes how we require the far end to log in on this channel. Requiring the other end to log in most frequently happens when they dial us (rather than the other way round), so this is likely to be one of several channels which are set using *svc listen*. Because of this, exact names and passwords are not attached to individual channels but are matched to particular users, as defined using the *user* command.

This command specifies that when using this channel, the user must log on using the specified protocol, and that they must provide any name/password combination which has been defined for that protocol, using the *user* command.

To remove this information on a channel, call *theylogin* with a single argument of *none*.

Configuration saving saves this information. By default, no login is required.

K.21.3 See also

user on page 876.

K.22 <channel> tunnel <n> <tunnel protocol> <dial direction>

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

K.22.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

K.22.2 Syntax

```
<channel> tunnel <n> <tunnel protocol> <dial direction>
```

K.22.3 Description

Logically associate the specified channel with the specified PPTP tunnel.

A single PPP channel can only be associated with a single interface, or a single tunnel.

Use *info* to find the current setting.

Calling with n=0 removes any association. This is the default state.

Configuration saving saves this information.

The possible tunnel protocols are: *pptp* and *l2tp*.

The *dial direction* may be: *in* or *out* for dial-in or dial-out respectively.

K.22.4 Example

```
ppp 3 tunnel 1 pptp out
```

K.22.5 See also

<channel> info on page 858, *<channel> interface* on page 859.

K.23 <channel> welogin

Note - This console command **has** been replaced by the CLI command *pppoa set transport welogin* on page 376. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

K.23.1 Syntax

```
<channel> welogin <name> <password> [pap|chap]

<channel> welogin none
```

K.23.2 Description

This command describes how we should log in to the far end when a connection is established. A name and password are supplied, and whether these should be used with the PAP or CHAP authentication protocol. CHAP is the default.

To remove this information on a channel, call *welogin* with a single argument of *none*.

If `chap` is specified, we will also log in using `pap` if the other end prefers this. If `pap` is specified we will only log in using `pap`.

Configuration saving saves this information. By default, no login is performed.

K.24 bcp

Note - This console command **has** been replaced by the CLI command *pppoa set transport headers hdlc* on page 344. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

K.24.1 Syntax

```
bcp stp|nostp
```

K.24.2 Description

This command describes parameters for BCP, the Bridge Control Protocol, which is used to transport MAC (Ethernet) packets over the PPP link. See the section entitled *Standards Conformance* in the *ATMOS PPP Functional Specification, DO-007078-PS* for a definition of the BCP option settings which are not controllable.

If *stp* is specified, the Spanning Tree Protocol is in use by the Bridges, to control bridge loops. In this case STP frames should be carried over any links using BCP.

If *nostp* is specified, STP frames should not be carried.

Configuration saving saves this information. By default, STP is not supported.

K.25 interface <n> localip

Note - This console command **has** been replaced by the CLI command *pppoa set transport localip* on page 354. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

K.25.1 Syntax

```
interface <n> localip <address>
```

K.25.2 Description

This command describes parameters for IPCP, the IP Control Protocol, when providing the server end of an IPCP connection. The server knows its own IP address (and may allocate an IP address to the remote end). This command tells the PPP process, for a particular interface, the local IP address to be associated with the local end.

For interface 1, this should be the same IP address as possessed by the *device ppp_device* in the IP stack. See the *IP dial-in server setup* on page 848 for more information.

If PPP channels are now associated with this interface, remote users can dial in to those channels and will be connected to the IP stack. They can be allocated IP addresses, see the command *<channel> remoteip* on page 867.

Call with *0.0.0.0* to remove any IP address setting. This is the default state.

Configuration saving saves this information.

K.25.3 See also

<channel> remoteip on page 867.

K.26 interface <n> stats

Note - This console command **has** been replaced by the CLI command *pppoa show transport* on page 378. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

K.26.1 Syntax

```
interface <n> stats
```

K.26.2 Description

The interface is regarded by the operating system as an Ethernet-like device which can be attached to the bridge or router, like other Ethernet devices in ATMOS. It also provides an ifEntry to SNMP providing basic information about traffic through the interface.

This command shows the basic information about byte and packet traffic through the interface, in SNMP terms.

K.27 user

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

K.27.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

K.27.2 Syntax

```
user add <name> [pwd <passwd> [pap|chap]]
user [<name>]
user delete <name>|all
```

K.27.3 Description

This command stores information about a particular login name/password combination. This is referred to as a *user*, regardless of whether it represents an individual.

When *user* is called on its own, information about all existing users is listed. When *user* <*name*> is called with no further arguments, details of that user alone are printed. Passwords are not shown.

Use *user delete* to delete an individual user by name, or to delete all users.

Use *user add* <*name*> to create a new user or update an existing one. The password is stored, and the authentication protocol which must be used for this user.

If a user is deleted or changed, existing sessions are not affected.

Configuration saving saves this information.

K.28 version

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

K.28.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

K.28.2 Syntax

```
version
```

K.28.3 Description

Provide the version number for the source of the *ppp* process.

L: PPTP Console Commands

This chapter describes the PPTP (Point-to-Point Tunnelling Protocol) Console commands.

L.1 Console object types

The PPTP process provides a number of PPTP connection tunnels. A tunnel consists of a control connection between the local PAC and a PNS, and a data connection (known as a call) through which a number of PPP connections or channels may be multiplexed.

The upper limits of several parameters may be configured using the *config resource* console command. These are listed in the section entitled *Resources and limits* in the *ATMOS PPTP Functional Specification, DO-007352-PS*.

The current state of each tunnel is saved by *config save*.

L.1.1 Console examples

These examples are for configuration of the PPTP Access Concentrator (PAC). Obviously the PPP client or server and the PNS must also be configured.

Dial-Out

The PPTP process uses functionality provided by the PPP process. Configure PPP channel 2 for an outgoing PPTP connection, using PPTP tunnel 1, and using PVC 800.

```
ppp 2 pvc 800
ppp 2 interface 0
ppp 2 tunnel 1 pptp out
ppp 2 enable
```

Next, configure the PPTP process to bind to an Ethernet interface with an IP address of, for example 192.168.10.1, and set up tunnel 1 to listen (waiting for the PNS to initiate the connection):

```
pptp bind 192.168.10.1
pptp 1 create listen
```

Dial-In

The PPTP process uses functionality provided by the PPP process. Configure PPP channel 2 for an incoming PPTP connection, using PPTP tunnel 1, and using PVC 800.

```
ppp 2 pvc 800 listen
```

```
ppp 2 interface 0
ppp 2 tunnel 1 pptp in
ppp 2 enable
```

Next, configure the PPTP process to bind to an Ethernet interface with an IP address, for example 192.168.10.1, and set up tunnel 1 with the PAC initiating the connection: to a PNS with IP address, for example, 192.168.10.2

```
pptp 1 bind 192.168.10.1
pptp 1 create 192.168.10.2
```

The rest of this section details the individual console commands provided.

L.2 bind

Note - This console command **has** been replaced by the CLI command *pptp set listening* on page 511. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

L.2.1 Syntax:

```
bind <ipaddress>|any|none
```

L.2.2 Description:

Specify which local interface to bind a listener to for incoming control connections.

If *ipaddress* is specified, PPTP will listen on port 1723 on that interface only for incoming control connections. Typically this will be the IP address of the local side network interface.

If any is specified, PPTP will accept control connections on any interface.

If none is specified, no incoming control connections will be accepted; in this case, tunnels may only be established via the local create and connect commands.

Configuration saving saves this information. The default is none.

L.2.3 Example

To listen for incoming control connections on local interface 192.168.1.1 only, enter:

```
pptp bind 192.168.1.1
```

L.2.4 See also

<tunnel> create on page 884, using the *listen* option.

L.2.5 Notes

An incoming connection can only be accepted if the listener has a free tunnel object allocated to it. (Such objects are allocated with the *<tunnel> create listen* command.) The tunnel object used will be freed for use again when the tunnel is closed by either end.

L.3 <tunnel> connect

Note - This console command **has** been replaced by the CLI command *pptp attach* on page 505. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

L.3.1 Syntax

```
<tunnel> connect
```

L.3.2 Description:

Explicitly connect a tunnel (that was created using *create* <*ipaddress*>) to the remote PNS that *create* specified, establishing the control connection.

L.3.3 Example

To connect tunnel 1 to configured PNS, enter:

```
pptp 1 connect
```

L.3.4 See also

<tunnel> create on page 884, using the <*ipaddress*> option, *<tunnel> disconnect* on page 886.

L.3.5 Notes

This command is meaningless if applied to a tunnel object that is allocated to the listener (as created with the <*tunnel*> *create listen* command); in this case it will produce an error message.

L.4 <tunnel> create

Note - This console command **has** been replaced by the CLI command *pptp add tunnel* on page 504. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

L.4.1 Syntax

```
<tunnel> create <ipaddress>|listen
```

L.4.2 Description:

Create a tunnel object.

If *ipaddress* is specified, the tunnel is associated with a remote PNS at that IP address. The control connection is not actually established until use of the tunnel is requested by PPP, or an explicit connect is issued.

If listen is specified, the tunnel is allocated for use by an incoming control connection from a remote PNS. At least one such tunnel must exist if any incoming connections are to be accepted at all.

Incoming connections are mapped to the first available listening tunnel object. It is not currently possible to use properties of the incoming connection (such as its IP address, or information supplied in the fields of the PPTP control messages) to map the connection to a specific tunnel.

Configuration saving saves this information. By default, no tunnels are created.

L.4.3 Example

To connect Tunnel 1 to PNS at 192.168.1.2, enter:

```
ptp 1 create 192.168.1.2
```

L.4.4 See also

<tunnel> connect on page 883, *<tunnel> delete* on page 885.

L.5 <tunnel> delete

Note - This console command **has** been replaced by the CLI command *pptp delete tunnel* on page 508. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

L.5.1 Syntax

```
<tunnel> delete
```

L.5.2 Description

Delete a tunnel object (the opposite of create). If the tunnel is currently connected, any active data connections across the tunnel are terminated and the control connection is closed.

L.5.3 Example

To delete Tunnel 1, enter:

```
pptp 1 delete
```

L.5.4 See also

<tunnel> create on page 884, using the <*ipaddress*> option.

L.6 <tunnel> disconnect

Note - This console command **has** been replaced by the CLI command *pptp detach* on page 509. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

L.6.1 Syntax

```
<tunnel> disconnect
```

L.6.2 Description

Explicitly disconnect a tunnel (the opposite of *connect*). All data connections across the tunnel are terminated and the control connection is closed.

If the tunnel object is associated with a particular remote PNS (as created with <*tunnel*> *create* <*ipaddress*>), it may be reconnected later, either explicitly with another connect command, or implicitly by PPP requesting to use it.

If the tunnel object is allocated to the listener (as created with <*tunnel*> *create listen*), it is freed for use by future incoming connections.

L.6.3 Example

To disconnect Tunnel 1, enter:

```
pptp 1 disconnect
```

L.6.4 See also

<tunnel> create on page 884, *<tunnel> connect* on page 883.

L.7 <tunnel> event

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

L.7.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

L.7.2 Syntax

```
<tunnel> event [<n>]
```

L.7.3 Description

Read or set the trace output level for a tunnel.

Configuration saving does not save this value. The default event level is 1; only very serious errors are reported.

The Event levels are listed in the table below:

Level	Description
1	Only very serious errors reported (default)
2	Definite protocol errors or very significant events reported.
3	Channels going up/down reported.
4	Every packet and significant state change is reported.
5	Every packet sent/received is disassembled, and hex dumped.

L.8 <tunnel> info

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

L.8.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

L.8.2 Syntax

```
<tunnel> info [all]
```

L.8.3 Description

Provide information about the current settings of this tunnel. This includes all configured state, and also current protocol information.

Specifying *all* prints out more information.

info and *status* are synonyms.

L.9 list

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

L.9.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

L.9.2 Syntax

```
list
```

L.9.3 Description

Lists all currently created tunnel objects and the IP address of the remote PNS associated with each one.

L.10 version

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

L.10.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

L.10.2 Syntax

```
version
```

L.10.3 Description

Provide the version number for the source of the pptp process.

M:TFTP Console Commands

This chapter describes the TFTP Console commands.

M.1 connect

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

M.1.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

M.1.2 Syntax

```
connect <node_name> || <ipaddress>
```

M.1.3 Scope:

Client mode only.

M.1.4 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.

M.1.5 Example

```
connect 192.168.200.10
```

M.1.6 See also

put on page 898, *get* on page 894.

M.2 get

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

M.2.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

M.2.2 Syntax

```
get <remote_file> [local_file]
```

M.2.3 Scope:

Client mode only.

M.2.4 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 (currently 8K) can be retrieved. This means that it 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.

M.2.5 Example

```
get ipaddresses
```

M.2.6 See also

connect on page 892, *put* on page 898.

M.3 help

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

M.3.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

M.3.2 Syntax

```
help
```

M.3.3 Description

The *help* command displays the help text which lists the (commonly used) TFTP commands. The *init* command is not listed in the help text.

The *trace* command has a large number of optional parameters and detail on this command may be displayed by typing *trace help*.

If the software build supports client mode operation, these commands will be displayed in the help text.

M.3.4 Example

```
help
```

M.3.5 See also

version on page 900, *trace* on page 899.

M.4 init

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

M.4.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

M.4.2 Syntax

```
init
```

M.4.3 Description

The *init* command causes all sessions to be initialised to an idle state. This command can be used during testing but is not required in normal operation. The command does not appear in the help text.

M.4.4 Example

```
init
```

M.5 list

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

M.5.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

M.5.2 Syntax

```
list
```

M.5.3 Description

The *list* command displays the status of any active sessions. This command is primarily intended for use during debug.

M.5.4 Example

```
list
```

M.6 put

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

M.6.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

M.6.2 Syntax

```
put [local_file] <remote_file>
```

M.6.3 Scope:

Client mode only.

M.6.4 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.

M.6.5 Example

```
put foo.txt
```

M.6.6 See also

connect on page 892, *get* on page 894.

M.7 trace

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

M.7.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

M.7.2 Syntax

```
trace <help> || <-*> || <+event1> <-event2>
```

M.7.3 Description

The *trace* command allows the user to examine the currently set trace types or add/subtract trace types. Trace help lists all the available tracing types.

If the *trace* command is used with no parameters, the currently set trace types are displayed.

M.7.4 Example

```
trace +tmr_exp
```

M.8 version

Note - This console command **has not** been replaced by a CLI command. If used, this command will create inconsistencies between the Information Model and the underlying system. It is a *blacklisted* console command that can only be used **with extreme caution**.

M.8.1 CLI access permission - blacklisted commands

Users with the following access permissions can use the blacklisted commands:

- Super user

M.8.2 Syntax

```
version
```

M.8.3 Description

The *version* command displays software version information about the process.

The version number, which is displayed in the form *a.bc*, is defined in the module file as an integer *abc*.

M.8.4 Example

```
version
```

M.8.5 See also

help on page 895.

N:Webserver Console Commands

This chapter describes the Webserver Console Commands.

N.1 webserver archive

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

N.1.1 CLI access permission - usable commands

Users with the following access permissions can use this command:

- Engineer
- Super user

N.1.2 Syntax

```
webserver archive <archive_filename>
```

N.1.3 Description

This command sets <*archive_filename*> to be the location of the derived archive file that will be used for static Web page content. This derived archive file will be loaded when the webserver process starts.

If no parameter is specified, the current value will be displayed.

If a value of *none* is specified, a derived archive file will not be loaded at startup.

Configuration saving saves this information.

Changes will only take effect after the configuration is saved and the system is restarted.

N.1.4 Example

To specify a derived archive contained in a binary file of *derived_data.dat* use the following syntax:

```
10.0.0.1 webserver> archive //isfs/derived_data.dat
```

N.1.5 See also

webserver load on page 906

N.2 webserver clearstats

Note - This console command **has** been replaced by the CLI command *webserver clear stats* on page 696. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.Syntax

```
webserver clearstats
```

N.2.1 Description

This command sets the *webserver* process counters to 0.

N.2.2 Example

```
10.0.0.1 webserver> clearstats
```

N.2.3 See also

webserver stats on page 909

N.3 webserver enable/disable

Note - This console command **has** been replaced by the CLI command *webserver enable|disable* on page 697. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

N.3.1 Syntax

```
webserver [enable | disable]
```

N.3.2 Description

This command enables/disables the webserver process.

N.3.3 Example

```
prompt> webserver enable
```

N.4 webserver help

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

N.4.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

N.4.2 Syntax

```
webserver help [all | <command>]
```

N.4.3 Description

Displays help about available console commands.

N.4.4 Example

```
10.0.0.1 webserver> help

Commands are:

version         status          enable          disable
port            archive         load            mgmtip
stats           clearstats

   '.'        repeats the last command
Type 'help all' or 'help <command>' for more details
```

N.5 webserver load

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

N.5.1 CLI access permission - usable commands

Users with the following access permissions can use this command:

- Engineer
- Super user

N.5.2 Syntax

```
webserver load <derived_archive_filename>
```

N.5.3 Description

This command loads a new derived archive file specified by <*derived_archive_filename*>.

If no parameter is specified, the name o the derived archive currently loaded will be displayed.

If a derived archive is currently loaded, it will be unloaded (removed from use) and the memory will be freed.

It is not necessary to use the *load* command if you have already specified a derived archive file using the *archive* command.

N.5.4 Example

To load a derived archive contained in a binary file of *derived_data.dat* use the following syntax:

```
10.0.0.1 webserver> load //isfs/derived_data.dat
```

N.5.5 See also

webserver archive on page 902

N.6 webserver mgmtip

Note - This console command **has** been replaced by the CLI command *webserver set managementip* on page 699. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

N.6.1 Syntax

```
webserver help [<IP_address> | any]
```

N.6.2 Description

This command allows connection requests to be restricted to only one IP address, e.g., from an IP address that is used by a management entity. If no parameter is specified, the current value will be displayed.

If a value of *any* is specified, requests from any IP address will be allowed.

Configuration saving saves this information.

Changes will only take effect after the configuration is saved and the system is restarted.

N.6.3 Example

To specify a management IP address use the following syntax:

```
10.0.0.1 webserver> mgmtip 10.0.0.2
```

To allow connections from any IP address, use the following syntax:

```
10.0.0.1 webserver> mgmtip any
```

N.7 webserver port

Note - This console command **has** been replaced by the CLI command *webserver set port* on page 700. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

N.7.1 Syntax

```
webserver port <http_port>
```

N.7.2 Description

This command sets the HTTP port number that the Web Server will use.

If no parameter is specified, the current value will be displayed.

Configuration saving saves this information.

Changes will only take effect after the configuration is saved and the system is restarted.

The default HTTP port number is 80.

N.7.3 Example

```
10.0.0.1 webserver> port 82
```

N.8 webserver stats

Note - This console command **has** been replaced by the CLI command *webserver show info* on page 702. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

N.8.1 Syntax

```
webserver stats
```

N.8.2 Description

Displays counters about the *webserver* process.

N.8.3 Example

```
10.0.0.1 webserver> stats

Web Server statistics:

Bytes transmitted: 38984

Bytes received:    617
```

N.8.4 See also

webserver clearstats on page 903

N.9 webserver status

Note - This console command **has** been replaced by the CLI command *webserver show stats* on page 703. Use the CLI command instead of the console command. Users with any of the three CLI user access settings (Default, Engineer or Super user) can use this CLI command.

N.9.1 Syntax

```
webserver status
```

N.9.2 Description

This command shows the status of the Web Server. The status information includes the enable state of the Web Server, the HTTP port number, and the derived archive file name.

N.9.3 Example

```
10.0.0.1 webserver> status

WebServer is enabled.

The HTTP port is 80.

WebServer archive file is //isfs/derived_data.dat.
```

N.10 webserver version

Note - This console command **has not** been replaced by a CLI command. If used, this command will not create inconsistencies between the Information Model and the underlying system. It is a *usable* console command.

N.10.1 CLI access permission - usable commands

Users with the following access permissions can use the usable commands:

- Engineer
- Super user

N.10.2 Syntax

```
webserver version
```

N.10.3 Description

This command displays the version number of the Web Server support component as well as the release number of Web Server core code.

N.10.4 Example

```
10.0.0.1 webserver> version

WebServer Version is 00.01 [Core_R5_2_4]
```

Index

Symbols

A

B

C

D

E

F

G

H

I

L

M

N

P

R

S

T

U

V

W