NOKIA

CLI Reference Guide for Nokia IPSO 4.2

Part No. N450000360 Rev 005

Published November 2008

080101

Nokia Contact Information

Corporate Headquarters

Web Site	http://www.nokia.com
Telephone	1 914 368 0400
Mail Address	Nokia Inc. 102 Corporate Park Drive White Plains, NY 10604 USA

Regional Contact Information

Americas	Nokia Inc. 102 Corporate Park Drive White Plains, NY 10604 USA	Tel: 1 877 997 9199 E-mail: usa@nokiaforbusiness.com
Europe, Middle East, and Africa	Nokia House, Summit Avenue Southwood, Farnborough Hampshire GU14 ONG UK	Tel: (UK) 44 161 601 8908 Tel: (France) 33 170 708 166 Tel: (Middle East, Africa, Dubai) 971 4 3697600 E-mail: europe@nokiaforbusiness.com E-mail: mea@nokiaforbusiness.com
Asia-Pacific	438B Alexandra Road #07-00 Alexandra Technopark Singapore 119968	Tel: 603 9145 1032 E-mail: asia@nokiaforbusiness.com

Nokia Global Technical Assistance Center

Web Site	https://support.nokia.com	
Voice	Americas	1 888 361 5030
	Europe, Middle East, Africa	44 1252 868900
	Asia-Pacific	65 6723 2999
	International	1 613 271 6721

Non-Technical Support

For non-technical support issues, including your Nokia Support Agreement, licensing, and Web site access, use the following contact information:
E-mail: es.service@nokia.com

080919

Contents

About the CLI Reference Guide

This guide describes the commands that you can run from the command-line interface (CLI). You can use the CLI to configure and monitor IPSO systems.

The CLI complements Nokia Network Voyager, the Nokia web-based interface for IPSO systems, by allowing you to chose the interface you are most comfortable with. A few commands, specifically some of the routing commands, have no equivilent in Network Voyager.

Most tasks that you can accomplish with Network Voyager you can also do with the CLI. You can enter CLI commands individually and you can also create batch files of CLI commands to automate configuration tasks. You should have a fundamental knowledge of routing principals, security software, firewalls, and command-line interfaces on UNIX-based systems.

This guide provides all of the information you need to create and implement command-line interface (CLI) commands that are applicable to IPSO.

Document Organization

This guide is organized into the following chapters:

- Chapter 6, “SNMP Commands”
- Chapter 7, “IPv6 Commands”
- Chapter 8, “Network Security and Access Commands”
- Chapter 9, “Routing Commands”
- Chapter 10, “Traffic Management Commands”
- Chapter 11, “Monitoring Commands”

Document Conventions

The following sections provide document conventions used throughout this guide.

Cautions and Notes

Caution
Cautions indicate potential equipment damage, equipment malfunction, loss of performance, loss of data, or interruption of service.

Note
Notes provide information of special interest or recommendations.

Command Syntax Conventions

The notation conventions described below are used in the CLI command descriptions and related text.

Note
The Nokia CLI prompt is omitted from the examples shown in this guide.

Command Syntax Example 1

```
set clienv
        debug <0-5>
        echo-cmd <on | off>
        on-failure <stop | continue>
        output <pretty | structured | xml>
        rows integer
        syntax-check <on | off>

save clienv
```

Text you enter is shown as monospace font; for example, `set clienv`.

Each line that is indented under an earlier component of the command is an argument for that command; for example, `debug`, `echo-cmd`, `on-failure`, `output`, `rows`, and `syntax-check` are all arguments for the `set clienv` command.

If more than one choice is applicable in the command string, the alternative, mutually exclusive choices are surrounded by angle brackets (< >) and separated by vertical lines (|) or by a hyphen if the choices cover a range; for example, `<pretty | structured | xml>` and `<0–65535>`. If a default value is applicable, that value is shown underlined; for example, `pretty`.

If a phrase or term in the command syntax is italicized, then that term or phrase is a placeholder for an entry you select. In the above example, where the line reads `rows integer`, your actual entry might be `rows 5`.

Command Syntax Example 2

```
set interface log_if_name address ip_address
```

If one of more phrases or terms are surrounded by square brackets then the information inside the brackets is optional and might or might not be included in your use of the command.

Additional Documentation

For supporting documentation, see the following documents:

- *Nokia Network Voyager Reference Guide*, which is available on the Nokia customer support Web site and is also available from the Network Voyager navigation tree (if you install the IPSO documentation package).
- *Clustering Configuration Guide for Nokia IPSO*, which is available on the Nokia customer support Web site and is also available from the Network Voyager navigation tree (if you install the IPSO documentation package).

  This guide explains many details about how to implement IP clusters.
- *Getting Started Guide and Release Notes for Nokia IPSO*, which is available on the Nokia customer support Web site.

  This document contains descriptions of the new features for the current IPSO release, installation instructions, and known limitations.

1 Introducing the Command-Line Interface

This chapter describes the configuration, administration, and monitoring tasks you can perform using the Nokia IPSO command-line interface (CLI).

To use the CLI:

1. Log on to the platform using a command-line connection (SSH, console, or telnet) over a TCP/IP network as an admin, cadmin, or monitor user.

   If you log in as a cadmin (cluster administrator) user, you can change and view configuration settings on all the cluster nodes. See Chapter 10, "Traffic Management Commands" for information about administering a cluster.

   If you log in as a monitor user, you can execute only the `show` form of commands. That is, you can view configuration settings, but you cannot change them.

2. Invoke the CLI using the one of the procedures explained in the next section.

> **Note**
> Nokia recommends that you press q instead of Ctrl-C to return to the CLI prompt. Under certain circumstances, entering Ctrl-C repeatedly might result in the system dumping a core file and exiting the CLI.
> If this occurs and there are configuration changes that you have not saved that you want to save, restart the CLI by entering **`clish`** and then entering **`save config`** at the CLI prompt.

Invoking the CLI

You can execute CLI commands from the CLI shell and the IPSO shell. Most users have the CLI shell as their default shell. However, the admin user has the IPSO shell (C shell) as their default shell.

Execute From	To Implement	Purpose
IPSO shell	Enter `clish` to invoke the CLI shell. The prompt changes, and you can then enter CLI commands.	Lets you enter any CLI commands in an interactive mode with help text and other helpful CLI features.
IPSO shell	Enter `clish -c "`*`cli_command`*`"`	Lets you execute a single CLI command. You must place double-quotation marks around the CLI command
Command files	• Enter `clish -f` *`filename`* • Enter `clish` to invoke the shell. Then enter `load commands` *`filename`*	Lets you load commands from a file that contains commands. The argument must be the name of a regular file.

IPSO Shell Options for CLI Commands

In addition to the `-c` and `-f` options, the IPSO shell supports the following command-line options:

Option	Purpose
-d	Sets the debug level; Enter an integer from 0 to 5 as the parameter
-o	Sets output format; enter either *`pretty`*, *`structured`*, or *`xml`*.
-s	Use with `-c` or `-f` to force a permanent configuration save. For example, enter `clish -s -f` *`filename`* or `clish -s -c "`*`cli_command`*`"`
-i	Use before `-f` option to continue loading commands from a file even if a command within the file fails.

Environment Commands

Use the following commands to set an environment for a particular session to modify the `.cshrc` file to set the environment permanently:

```
set clienv
        debug <0-5>
        echo-cmd <on | off>
        on-failure <stop | continue>
        output <pretty | structured | xml>
        prompt name
        rows integer
        syntax-check <on | off>

save clienv
```

Arguments

`debug <0-5>`	Specifies the debug level. Level 0 specifies not to perform any debugging, to display error messages only. Level 5 specifies the highest level of debugging **Default:** 0
`echo-cmd <on | off>`	Specifies to echo all commands. When using the `load commands` command, all commands are echoed before being executed. **Default:** off
`on-failure <stop | continue>`	Continue specifies to continue executing commands from a file or a script and only to display error messages. Stop specifies to stop executing commands from a file or a script when the system encounters an error. **Default:** stop
`output <pretty | structured | xml>`	Specifies the command-line output format. See "Output Formats" on page 27 for more detailed information. **Default:** pretty
`prompt` *`name`*	Specifies the appearance of the command prompt. To set the prompt back to the default, use the keyword `default`.
`rows` *`integer`*	Specifies the number of rows to display on your console. **Default:** Specified by your console or xterm window.

`syntax-check <on | off>`	Specifes to put the shell into syntax-check only mode. Commands you enter are checked syntactically and are not executed, but values are validated. **Default:** off
`save clienv`	Specifies to save the environment variables that the user modifies with the `set clienv` commands.

Use the following commands to view the environment settings on your system.

```
show clienv
        debug
        echo-cmd
        output
        on-failure
        output
        rows
        syntax-check
```

Arguments

`debug`	Displays the configured debug level.
`echo-cmd`	Displays whether or not echo-cmd is enabled.
`on-failure`	Displays whether or not on-failure is enabled.
`output`	Displays the configured output.
`rows`	Displays the number of screen rows configured.
`syntax-check`	Displays whether or not syntax-check is enabled.

Transaction Mode

You can use transaction mode to enter a series of CLI commands that are executed as a group. This mode is particularly useful if you want to use configuration scripts and don't want to commit changes to the configuration database unless all the commands in the group are executed successfully.

When transaction mode is active, you can enter as many CLI commands as you want. The commands are executed but not committed to the configuration database, and you see an error message if a command fails. You can have a script look for error messages and roll back (undo) all the changes if it detects any errors.

To start transaction mode, enter

```
start transaction
```

`[Xact]` is appended to the prompt to let you know that transaction mode is active.

Enter changes that should be implemented as a group.

To implement changes made in transaction mode and commit them to the configuration database, enter

```
commit
```

To roll back the changes you just made in transaction mode, enter

```
rollback
```

After you enter `commit` or `rollback`, the CLI leaves transaction mode.

Output Formats

CLI supports three output formats: pretty, structured, and xml. Use the -o option at the command line to set one of the supported formats, except for pretty output, which is the default. For example, to enter the CLI shell and print output in xml format, enter `clish -o xml` from the IPSO shell.

The pretty mode generates output, as in the following example:

```
User admin
   gid 0
   home /var/admin
   passwd $1$_J9..w8j$yBA/JaVED1rk2DiPm1XHF
   realname Admin
   shell /bin/csh
   uid 0
```

The structured mode generates output, as in the following example:

```
User;Admin;
gid;0
home;/var/admin;
passwd;$1$_J9..w8j$7BA/JaVED1rk2DiPm1XHF/;
realname;Admin;
shell;bin/csh;
uid;0;
```

The xml mode generates output that is embedded in xml, as in the following example:

```
<user> admin
   <gid>0</gid>
   <home>/var/admin</home>
.
.
.
</user>
```

General CLI Features

This section describes general CLI features.

Commands and Command Operations

A command always starts with a operation, such as `set` or `add`, followed by a feature, such as vrrp, followed by one or more arguments, such as accept-connections. The possible operations are:

- add—adds a new value to the system.
- commit—ends transaction by committing changes.
- delete—removes a value from the system.
- download—downloads an IPSO image
- exit—exits from the CLI or IPSO shell.
- halt—halts the system.
- load—loads commands from a file.
- quit—exits from the CLI.
- reboot—reboot the system.
- rollback—ends transaction by discarding changes.
- save—saves the configuration changes made since the last save.
- set—sets a value in the system.
- show—displays a value or values from the system.
- start—starts transactions.
- upgrade—upgrades packages
- ver—displays the version of the active IPSO image.

Command Completion

Press Enter to execute a finished command string. The cursor does not have to be at the end of the line when you press Enter. You can usually abbreviate the command to the smallest number of unambiguous characters.

Using Tab to Expand Commands

The Tab key provides two methods of automatic command-line completion.

- If you enter the main keyword for a command, such as vrrp as in the example below, press Space, and then press Tab, the console displays the initial arguments that the command for that feature accepts. After the initial argument display, the command prompt and the command you originally entered are displayed.

  For example,

  ```
  Nokia> set vrrp <Space><Tab>
  accept-connections - Accept-connections
  coldstart-delay - Coldstart-Delay
  interface - Interface
  Nokia> set vrrp
  ```

- If you enter the feature keyword and part of an argument and press Tab (without pressing Space), the console displays the possible arguments that match the characters you typed. command option for that argument only. In this case, the console does not display all the command arguments.

  For example,

  ```
  Nokia> set in<Tab>
  inatmarp  - Set the parameters which regulate Inverse ATM ARP
  protocol behavior
  interface - Configures the interface related parameters
  ```

In either case, pressing Tab causes the CLI to display possible values for the next argument only. The CLI does not indicate what arguments (if any) can be typed after the next argument.

Using Esc to Expand Commands

You can use Esc to see all the possible arguments that could be used to complete a command. To use this form of command completion, enter a partial command and then press Esc twice, as shown in the following example.

```
Nokia> set in<Esc><Esc>
set inatmarp holdoff-time VALUE
set inatmarp keep-time VALUE
set inatmarp max-retries VALUE
set inatmarp timeout VALUE
set interface VALUE [ vlanid VALUE logical-name VALUE comments VALUE ]
set interface VALUE [ vlanid VALUE logical-name VALUE disable enable ]
set interface VALUE logical-name VALUE
set interface VALUE status VALUE
set interface VALUE vc-max VALUE
 .
 .
 .
```

Viewing Related Commands

Use the following command to display all the available commands for a combination of operation and feature.

```
show commands [ op <value> ] [ feature <value>]
```

Arguments

`op <value>`	Displays commands for the particular operation you enter. The range is show, set add, and delete.
`feature <value>`	Displays commands for the specific feature you enter, for example, bgp or snmp.

For example, if you enter

```
show commands op set feature interface
```

the system responds

```
set interface VALUE [ vlanid VALUE logical-name VALUE comments VALUE ]
set interface VALUE [ vlanid VALUE logical-name VALUE disable enable ]
set interface VALUE logical-name VALUE
set interface VALUE status VALUE
set interface VALUE vc-max VALUE
set interface VALUE vcs VALUE
.
.
.
```

You can also omit specifying an operation. If you do so, the system displays all of the commands that are valid for the specified feature. For example, if you enter

```
show commands feature interface
```

the system lists all of the commands that you can use to manage interfaces.

Using Default Values

Some values are in effect by default. If you change one of these to something other than the default, you can change it back by using the argument `default`.

For example, the default ARP keep-time value is 14400 seconds. If you had set the keep-time value to something else, you could reset it to 14400 seconds by entering

```
set arp keep-time default
```

Using the argument `default` is a convenient way to configure the system to use standard values without having to know what the values are .

In this document, default values are shown underlined. For example, the default speed of ethernet interfaces is 10 megabits per second, and this is shown in the syntax example like this:

```
speed <10M | 100M | 1000M>
```

In some cases, default values is are not indicated in syntax examples. For example, the range of valid ARP keep-time values is 1–86400 seconds, so the relevant syntax example is shown like this:

```
keep-time <1–86400>
```

The accompanying text notes that the default keep-time value is 14400 seconds.

Command Help

If you enter a command or part of a command and enter a question mark (?), the console displays help on that command, keyword, or value. This help feature is not available for routing commands.

For example:

```
Nokia> set ipsec?
Commands to configure IPsec.

Nokia> set ipsec log-level?
Verbosity of the logs generated.
Can be ERROR, DEBUG or INFO. Default value is ERROR
```

Command Recall

You can recall commands using the up and down arrow keys, similar to the UNIX Bash shell. The up arrow first recalls the last command, the next to last command, and so on.

Executing Previous Commands

The following list shows the history commands you can enter that execute complete commands:

- `history`–displays the last 100 commands.
- `!!`—executes the most recent command.
- `!`*`nn`*—in which *`nn`* is the number of a specific command from the history list, executes a previous command.
- `!-`*`nn`*—in which *`nn`* is the nnth previous command. For example, entering `!-3` executes the third from the last command
- `!`*`str`* —executes the most recent command starts with *`str`*.
- `!\?`*`str`*`\?`–executesthe most recent command containing *`str`*. The trailing `?` may be omitted if *`str`* is followed immediately by a new line.
- `!!:s/`*`str1`*`/`*`str2`* —repeats the last command, replacing *`str1`* with *`str2`*.

Reusing Parts of Commands

You can combine word designators with history commands to refer to specific words used in previous commands. Words are numbered from the beginning of the line with the first word being denoted by `0`. Use a colon to separate a history command from a word designator. For example, you could enter `!!:1` to refer to the first argument in the previous command. In the command `show interfaces`, `interfaces` is word 1.

- `0`—The operation word.
- `n`—The nth word.
- `^`—The first argument; that is, word 1.
- `$`–The last argument.
- `%`—The word matched by the most recent `\?`*`str`*`\?` search.

Immediately after word designators, you can add a sequence of one or more of the following modifiers, each preceded by a colon:

- `p`—Print the new command but do not execute.
- `s/`*`str1`*`/`*`str2`* —Substitute `new` for the first occurance of old in the word bieng referred to.
- `g`—Apply changes over the entire command. Use this modified in conjunction with `s`, as in `gs/`*`str1`*`/`*`str2`*.

Command-Line Movement and Editing

You can back up in a command you are typing to correct a mistake. To edit a command, use the left and right arrow keys to move around and the Backspace key to delete characters. You can enter commands that span more than one line.

The following list shows the keystroke combinations you can use:

- Alt-B—Go to the previous word.
- Alt-D—Delete next word.
- Alt-F—Go to the next word.
- Alt-Ctrl-H—Delete the previous word.
- Alt-Ctrl-L—Clear the screen and show the current line at the top of the screen.
- Alt-Ctrl-_ —Repeat the previous word.
- Ctrl-A—Move to the beginning of the line.
- Ctrl-B—Move to the previous character.
- Ctrl-E—Move to the end of the line.
- Ctrl-F—Move to the next character.
- Ctrl-H—Delete the previous character.
- Ctrl-L—Clear the screen and show the current line at the top of the screen.

- Ctrl-N—Next history item.
- Ctrl-P—Previous history item.
- Ctrl-R—Redisplay the current line.
- Ctrl-U—Delete the current line.

Exiting an Output Screen

When you enter a CLI command that produces more than one screen of output (such as `show route all`), the display stops scrolling when the window is full and the `-- More --` prompt is shown. To exit the output screen, enter q.

If you enter a number of commands such as these and repeatedly press Ctrl-C when the `-- More --` prompt is displayed, the system might dump a core file and exit from the CLI. If there are any configuration changes that you have not saved (and that you want to save), follow these steps:

1. Restart the CLI by entering **`clish`**.
2. At the CLI prompt enter

   `save config`

Setting Configuration Locks

When you launch the CLI shell, the shell attempts to acquire an exclusive configuration lock. If there is an active CLI or Voyager session that has already acquired an exclusive configuration lock, a message appears. You can execute show commands, but you cannot change any settings unless you override the configuration lock.

Use the following commands temporarily restrict the ability of other admin users to make configuration changes. This feature allows you to lock out other users for a specified period of time while you make configuration changes.

```
set config-lock
        <on | off>
        on timeout <5-900>
        on override
```

Arguments

`<on | off>`	Specifies whether to enable or disable configuration lock. When you enable config-lock, the default timeout value is 300 seconds. **Default:** off
`on timeout <5-900>`	Specifes to enable config-lock for the specified interval in seconds.
`on override`	Specifies to override an existing config-lock and thus disable config-lock.

Monitoring the File System and Processes

Use the following commands to monitor the system's file system and processes and to view memory capacity.

```
show fsinfo

show processes

show swapinfo
```

Arguments

`fsinfo`	Displays the number of file systems, the directories in which they are mounted, and their capacity.
`processes`	Displays the currently running processes

`swapinfo`	Displays the amount of memory available for swapping into the kernel.

Loading Commands From a File

You can execute a series of CLI commands from a text file. The file can contain only commands and comments. Each comment line must begin with the pound character (#). To split a command between multiple lines, type an escape character (\) at the end of each line. Do not type any characters, including spaces, after the escape character.

You can create and edit the file on the IPSO system using the VI text editor. You can also create the file on a remote system and copy the file to the IPSO system using FTP.

For example, you could create a file `foo.txt` that contains a series of CLI commands. To execute the commands in the file from the IPSO shell (not the CLI) you would enter:

```
IPSO[admin]# clish -f foo.txt
```

This assumes that `foo.txt` is in the `/var/admin` directory, which is the default directory for `admin`. If the command file is in a different directory or if you have changed to a different directory, modify the path accordingly.

The `-f` option allows the system to read commands from a file. You can also use the `-i` option to force the system to ignore errors in the results of the commands. The CLI normally stops reading commands from a file when a command fails.

You could execute the commands in `foo.txt` from the CLI by entering:

```
Nokia> load foo.txt
```

If you want the CLI to ignore errors in the results of commands and continue executing the commands in the file, enter the following command before loading the file:

```
Nokia> set clienv on-failure continue
```

Reset the CLI to stop on errors by entering:

```
Nokia> set clienv on-failure stop
```

Using IPSO Shell Commands

While using the CLI, you can start a standard shell that allows you to execute standard shell commands (such as `ping`, `traceroute`, and so on) by entering

```
shell
```

To exit this shell and return to the CLI, enter

```
exit
```

Saving Configuration Changes

Configuration changes you enter using the CLI are applied immediately to the running system. To ensure that these changes remain after you reboot, that is, to save your changes permanently, enter `save config` if you are using interactive mode. If you want to save your configuration changes into a different file**,** enter `save cfgfile` *`filename`***.**

If you use command-line mode and the `-c` option, you must use the `-s` option to save your configuration changes permanently. For example, enter:

```
clish -s -c "cli_command"
```

If you use the command-line mode and the `-f` option, you can use the `-s` option. For example, enter:

```
clish -s -f filename
```

If you use `-f`, you can also save your changes by including `save config` at the end of the file of configuration commands.

2 Interface Commands

This chapter describes the commands that you use to manage physical and logical interfaces network in your Nokia appliance.

General Commands

The commands described in this section apply to all the interfaces installed in the system.

Viewing All Interfaces

To see a variety of information about all the interfaces in a system, enter

```
show interfaces
```

Interface Names

When a physical interface is installed, the system automatically creates a corresponding logical interface and supplies default names for the physical and logical interface. To make an interface functional, you need to configure both the physical interface and at least one corresponding logical interface (you can create multiple logical interfaces for a single physical interface in some cases).

The `show interfaces` command displays the physical and logical names of all the installed interfaces (as well as other information). You use these names when viewing or configuring specific interfaces.

The following table explains the conventions used for interface names in this document.

if_name	Physical or logical interface name is acceptable.
phys_if_name	Only a physical interface name is acceptable. Physical interface names are assigned by the system and cannot be changed.
log_if_name	Only a logical interface name is acceptable. The default name for a logical interfaces is the name of the physical interface with *cunit_number* appended (in which *unit_number* uniquely identifies the logical interface). For example, the default name for the first logical interface created for physical Ethernet interface `eth-s1p1` is `eth-s1p1c0`. You can change the logical names of interfaces.

Deleting Any Logical Interface

On systems that support hot swapping of interfaces, removing a physical interface while the system is running will not cause any of its logical interfaces to be modified or deleted. If you reinstall the removed interface in the same slot, you do not have to reconfigure the logical interfaces.

If you permanently remove an interface, you may want to remove its configuration information. (For example, you may want to avoid seeing outdated information when you execute `show interfaces`.) To delete a logical interface, enter the following command.

```
delete interface log_if_name
```

To delete all the configuration information for a physical interface, enter the following command.

```
delete interface phys_if_name
```

To delete the IP address of a logical interface (without deleting the logical interface itself), enter the following command.

```
delete interface log_if_name address ip_address
```

If you delete all the logical interfaces or all the IP addresses for an interface, the interface will no longer be accessible over the network. If you delete all the logical interfaces or all the IP addresses for all the connected interfaces, the IP system will no longer be accessible over the network. If this occurs, restore network access to the system by connecting to it using a console connection and creating a logical interface for one of the connected physical interfaces. See the section in this chapter on the appropriate type of physical interface for information about how to do this.

Viewing Tunnels

To see information about all the VPN tunnels configured on a system, enter

```
show tunnels
```

Viewing Status and Statistics

To see if an interface is active, enter

```
show interface if_name status
```

To see various statistics about an interface, enter

```
show interface if_name statistics
```

To see the properties of and interface and whether the interface is active, enter

```
show interface if_name all
```

ARP

This section contains commands to configure the Address Resolution Protocol (ARP).

ARP Commands

Use the following commands to configure global ARP behavior.

```
set arp
        keep-time <60-86400>
        retry-limit <1-100>
        accept-multicast-replies <on | off>
        mirroring <on | off>
```

Use the following commands to show the current ARP settings.

```
show arp
        keep-time
        retry-limit
        accept-multicast-replies
        mirroring
        all
```

Arguments

`all`	Shows all the current configuration settings.
`keep-time <60–86400>`	Specifies or shows the number of seconds to keep resolved dynamic ARP entries. If an entry is not referred to and is not used by traffic before the time elapses, it is deleted (and the system will have to send a new request for the MAC address before it can send traffic to the interface). **Default:** 14400 seconds (4 hours).
`retry-limit <1–100>`	Specifies or shows the number of times to retry ARP requests (up to once per second) until holding off requests for the holdoff time (20 seconds). **Default:** 3
`accept-multicast-arpreplies <on | off>`	Specifies or shows whether the router accepts ARP replies with a multicast address. **Default:** off
`mirroring <on | off>`	Specifies or shows whether the VRRP-enabled interfaces on VRRP backup routers have the same ARP information as the master. Enabling this option can speed VRRP failovers because the new VRRP master does not need to learn the MAC addresses that correspond to its next hop IP addresses before it can forward traffic. **Default:** off

Use the following commands to add proxy and static ARP addresses.

```
add
        arpproxy address ip_address <macaddress mac_address | interface
           log_if_name>
        arpstatic address ip_address macaddress mac_address
```

`arpproxy address <macaddress` *`mac_address`* `| interface` *`log_if_name`*`>`	A proxy ARP entry makes this system respond to ARP requests for *`ip_address`* (usually an interface on another system) with *`mac_address`* or *`log_if_name`*. *`mac_address`* must be a valid MAC address (on this system) with six hexadecimal octets separated by colons. If you use the `interface` argument, *`log_if_name`* must be the logical name of an interface. (If the relevant physical interface has more than one logical interface, you must use the first logical interface .) If you use this argument, the system responds to ARP requests for *`ip_address`* with the MAC address of the interface specified by *`log_if_name`*. Proxy ARP entries will not be used when forwarding packets.

Use the following commands to show the current proxy, static, and dynamic ARP entries.

```
show arpproxy all

show arpstatic all

show arpdynamic all
```

Arguments

`arpproxy all`	Shows all the proxy ARP entries for the system.

`arpstatic all`	Shows all the static ARP entries for the system.
`arpdynamic all`	Shows all the dynamic ARP entries for the system.

Use the following commands to delete ARP addresses.

```
delete
        arpproxy address ip_address
        arpstatic address ip_address
```

ATM Interfaces

Use the commands explained in this section to configure physical and logical ATM interfaces.

Physical ATM Interfaces

Use the following commands to configure and view the settings of physical ATM interfaces.

```
set interface phys_if_name
        active <on | off>
        framing <sonet | sdh>
        transmitclock <freerun | looptiming>
        atm-oam <on | off>
        vc-max maxVPI/maxVCI
```

```
show interface phys_if_name
        all
        framing
        transmitclock
        atm-oam
        vc-max
        statistics
        lb-status
        status
```

Arguments

`all`	Shows all the current configuration settings.
`active <on | off>`	Specifies whether the interface is on or off. **Default:** on
`framing <sonet | sdh>`	Specifies the framing format used in this interface. The setting should match the type of transmission network this interface is connected to. **Default:** sonet
`transmitclock <freerun | looptiming>`	Specifies the clock source for the transmitted signal. The `freerun` argument selects the internal clock. If two ATM interfaces are connected directly with each other, at least one of them must use an internal clock. The `loop-timing` choice causes the interface to derive the transmit clock from the recovered receive clock. **Default:** freerun

`atm-oam <on | off>`	Specifies whether OAM cell processing is enabled at the ATM VC layer. If OAM is enabled, the interface sends LB responses and RDI cells to active VCs when needed. **Default:** off
`vc-max` *maxVPI*`/`*maxVCI*	Specifies or shows the ranges of virtual path identifiers (VPIs) and virtual channel identifiers (VCIs) for the ATM interface. The possible values are • `0/4095`: The VPI must be `0` and the range of possible VCIs is `1-4095`. • `1/2047`: The range of VPIs is `0-1` and the range of possible VCIs is `1-2047`. • `3/1023`: The range of VPIs is `0-3` and the range of possible VCIs is `1-1023`. • `255/15`: The range of VPIs is `0-255` and the range of possible VCIs is `1-15`. To see information about setting the actual VPI and VCI values, see page 48.
`statistics`	Shows traffic and error information about the interface.
`lb-status`	Shows the loopback status of virtual channels.
`status`	Shows whether the interface is active or inactive.

Logical ATM Interfaces

Use the following commands to create logical ATM interfaces.

```
add interface phys_if_name [unit <1—255>] type
        ipoa vcs [VPI/]VCI(s) [logical-name log_if_name]
        p2p vc [VPI/]VCI [logical-name log_if_name]
```

Arguments

`unit <1—255>`	Identifies a specific logical interface. The default name of the logical interface is the name of the physical interface followed by `c` and this number. For example, `entering add interface atm-s1p1 unit 0` creates the logical interface `atm-s1p1c0`.
`ipoa vcs [`*`VPI`*`/]`*`VCI(s)`*	Creates a logical interface that is part of an IP over ATM connection (also called a logical IP subnet, or LIS). You can specify a virtual path identifier (the default is `0`) and one or more virtual channel identifiers. The acceptable values for *`VPI`* and *`VCI(s)`* depend on the values you set using the command `set interface` *`phys_if_name`* `vc-max (see page 47)`. You can create multiple VCIs by specifying ranges (for example, `8-15`) or using commas to separate individual VCIs (for example, `8,11,15`). The following are additional examples of valid values for *`VCI(s)`* and *`VPI/VCI(s)`*: • `8` • `5,8-11` • `3/5,8` • `1/5-8,11,15-17`

p2p vc [*VPI*/]*VCI*	Creates a point-to-point logical interface. You can specify a virtual path identifier (the default is 0) and one virtual circuit identifier.The acceptable values for *VPI* and *VCI(s)* depend on the values you set using the command set interface *phys_if_name* vc-max (see page 47).
logical-name *log_if_name*	Specifies a logical name for the interface (which replaces the default logical name).

Use the following commands to configure logical ATM interfaces.

```
set interface log_if_name
        mtu <768—9180>
        address ip_address[/mask <8-30>]
        destination address
        unnumbered <yes | no>
        proxy-interface log_if_name
        enable | disable
        vcs [VPI/]VCI(s)
```

Arguments

mtu <768—9180>	Specifies the maximum transfer unit for the interface. The value must be an integer. **Default:** 1500
address *ip_address*[/ *mask* <8-31>]	Specifies a local IP address for the logical interface. For ATM over IP interfaces, you must include the subnet mask length of the IP subnet connected to the interface.

`destination` *`address`*	Specifies an IP address for the remote end of a point-to-point link—this should be the address of the interface at the other end of the link. This command is valid only for point-to-point interfaces.
`unnumbered <yes | no>`	Enables or disables unnumbered mode. In unnumbered mode the interface does not have its own unique IP address—the address of another interface is used. Specify the other interface with the `proxy-interface` *`log_if_name`* form of this command. This command is valid only for point-to-point interfaces.
`proxy-interface` *`log_if_name`*	Use when unnumbered mode is enabled to specify the interface from which the address for this interface is taken. *`log_if_name`* is the logical name of the other (numbered) interface. This command is valid only for point-to-point interfaces.
`enable | disable`	Enable or disable the logical interface.

`vcs [`*`VPI`*`/]`*`VCI(s)`*	Modifies the virtual path identifier and/or virtual channel identifiers of an IPoA interface. (You cannot modify the VPI or VCI of a point-to-point interface. If you want to change a point-to-point interface, delete the logical interface and create a new one with the appropriate settings.) For IPoA interfaces, you can create multiple VCIs by specifying ranges (for example, `8-15`) and/or using commas to separate individual VCIs (for example, `8,11,15`). The following are additional examples of valid values for *`VCI(s)`* and *`VPI/VCI(s)`* for IPoA interfaces: • `8` • `5,8-11` • `1/5-8,11,15-17`

Use the following command to view the settings for a logical ATM interface.

```
show interface log_if_name
        all
        status
        unnumbered
        address
        destination
        mtu
        proxy-interface
        vcs
        type
        statistics
```

Arguments

`all`	Shows all the current configuration settings.

`status`	Shows whether the interface is active or inactive.
`unnumbered`	Shows whether the interface is unnumbered.
`address`	Shows the local IP address configured on the logical interface, if any. This argument is valid only for numbered interfaces.
`instance`	Shows the routing instance that this address belongs to.
`destination`	For point-to-point interfaces, this field contains the IP address configured for the remote end of the link. For broadcast media, this field contains the network address and the network mask length of the IP subnet connected to the interface. This argument is valid only for numbered interfaces.
`mtu`	Shows the maximum transmission unit of the interface.
`proxy-interface`	For unnumbered interfaces, this shows the logical name of the numbered interface from which the address for this interface is taken. (If you have changed the logical name of the proxy interface from the default name, this command does not show you the new name—it shows the original (default) name of the interface.) Only point-to-point interfaces can be configured as unnumbered.
`vcs`	Shows the virtual channel identifier(s) of the interface (IPoA or point-to-point).

`type`	Shows the type of the interface—IP over ATM (`ipoa`) or point-to-point (`p2p`).
`statistics`	Shows traffic and error information for the interface.

ARP Entries for IPoA Interfaces

Use the following command to create ARP entries for IP over ATM logical interfaces.

add interface *`log_if_name`* atmarp vc [*`VPI`*/]*`VCI(s)`* remote *`ip_address`*

Static ARP entries map IP addresses to virtual channels (the combination of a VPI and VCIs). They can be created for an interface only if the interface has an IP address assigned to it.

For a given physical ATM interface, only one logical interface can be assigned a static ARP entry at a given time—that is, a physical ATM interface supports only one association of an IP address to a virtual channel at a time.

Arguments

`vc [`*`VPI`*`/]`*`VCI(s)`*	Specifies the virtual path identifier and one or more of virtual channel identifier(s) of the logical interface. (The default VPI value is 0.) These VPI and VCI values specified here must match values already assigned to the logical interface being proxied. The following are additional examples of valid values for *`VCI(s)`* and *`VPI/VCI(s)`*: • `8` • `5,8-11` • `1/5-8,11,15-17`

remote *ip_address*	Specifies the IP address of the remote end of the virtual channel. Packets received by *log_if_name* that are addressed to the remote IP address will be forwarded to it.

Use the following commands to view and delete ARP entries for IP over ATM logical interfaces.

```
show interface log_if_name atmarp
        static
        dynamic


delete interface log_if_name atmarp
        static vc VCI
        dynamic vc VCI
```

Arguments

`static`	Specifies a static ATM ARP entry (configured manually).
`dynamic`	Specifies a dynamic ATM ARP entry (configured by the system).
`vc` *VCI*	Specifies the virtual channel identifier of the ATM ARP entry.

Use the following commands to configure global InATMARP protocol settings.

```
set inatmarp
        keep-time <1-900>
        timeout <1-30>
        max-retries <1-100>
        holdoff-time <1-900>
```

The InATMARP protocol is used to resolve IP addresses to ATM addresses in a logical IP subnet (LIS) on top of an ATM network.

Use the following command to view all the global InATMARP protocol settings.

```
show inatmarp
```

Arguments

`keep-time <1-900>`	Specifies the number of seconds to keep resolved dynamic ATM ARP entries. **Default:** 900
`timeout <1-30>`	Specifies the InATMARP request retransmission interval in seconds. This value must be less than a third of the `keep-time` value . **Default:** 5
`max-retries <1-100>`	Specifies the number of times the system should retry InATMARP requests until holding off requests for the holdoff time. **Default:** 5
`holdoff-time <1-900>`	Specifies the number of seconds to hold off InATMARP requests after the maximum number of retries has been reached. **Default:** 60

Ethernet Interfaces

Use the commands explained in this section to configure physical and logical ethernet interfaces.

Note
Ethernet is the only interface type supported by the IP2250 and IP2255 platforms.

Physical Ethernet Interfaces

Use the following commands to configure and view the settings for physical Ethernet interfaces.

```
set interface phys_if_name
        speed <10M | 100M | 1000M>
        duplex <full | half>
        auto-advertise <on | off>
        link-recog-delay <1-255>
        active <on | off>
        flow-control <on | off>
        udld-enable <on | off>
        descriptor_size <128-512>


show interface phys_if_name
        speed
        duplex
        auto-advertise
        link-recog-delay
        flow-control
        status
        udld-enable
```

Arguments

`speed <10M | 100M | 1000M>`	Specifies the speed, in megabits per second, at which the interface will operate. **Default:** 10M

`duplex <full | half>`	Specifies the duplex mode in which the interface will operate. It must be the same as the port to which it is connected. For Gigabit Ethernet interfaces, this value must be full. **Default:** half
`auto-advertise <on | off>`	Specifies whether the interface will advertise its speed and duplex setting using Ethernet autonegotiation. This argument is not valid for Gigabit Ethernet interfaces. **Default:** on
`link-recog-delay <1-255>`	Specifies how many seconds a link must be before the system declares the interface is up. **Default:** 6
`flow-control <on | off>`	Specifies whether flow control is on. This argment is valid only for Gigabit Ethernet interfaces. **Default:** on
`active <on | off>`	Specifies whether the physical interface is active. **Default:** on
`status`	Shows whether the physical interface is active.
`udld-enable <on | off>`	Specifies whether to use the Cisco Unidirectional Link Detection (UDLD) protocol to improve detection of partial failures in fiber links. This argument is valid only for fiber-optic interfaces. You must enable UDLD on both ends of the link. **Default:** off

descriptor_size <128-512>	Specifies the number of descriptors that are available for Gigabit Ethernet interfaces. Increasing this value allows the system to temporarily store more packets while waiting for the CPU to service them. The system uses one descriptor per packet unless it receives jumbo frames (Ethernet frames larger than 1518 bytes), in which case it uses multiple descriptors per packet.The acceptable values are 128, 256, and 512. **Default:** 128

Logical Ethernet Interfaces

Use the following commands to create, configure, and view information about logical Ethernet interfaces.

```
add interface <log_if_name | phys_if_name> [vlanid <2—4094>] address
   ip_address/<0—31>
        comments comments
        logical-name new_log_if_name
        unit <1—4094>
        enable | disable

set interface log_if_name
        arp-mirroring <on | off>
        comments comments
        vlanid <2—4094>
        logical-name new_log_if_name
        enable | disable
        MTU <1500-16,000>
```

```
show interface log_if_name
        arp-mirroring
        comments
        vlanid
        logical-name
        MTU <1500-16,000>
```

Arguments

log_if_name | *phys_if_name*	When configuring the default logical interface, specify the logical name. This name ends with c0—for example, eth-s3p2c0. When adding a logical interface (in addition to the default logical interface), specify the physical interface. When adding a logical interface, you must specify a VLAN ID.
`unit <1—4094>`	Specifies the final digits of the logical name (the digits after the c) when adding a logical interface. If you do not specify the unit, IPSO creates the number.
`arp-mirroring <on | off>`	If VRRP is enabled on this interface, specifies whether it should learn the same ARP information as the master if is on a backup router. Enabling this option can speed VRRP failovers because the new VRRP master does not need to learn the MAC addresses that correspond to its next hop IP addresses before it can forward traffic
`comments` *comments*	Specifies comments about an interface. Bracket multiple word comments with quotation marks.

`vlanid <2–4094>`	Specifies the virtual LAN that the logical interface is assigned to. You cannot assign a virtual LAN ID to the first logical interface for a given physical interface.
`address` *`ip_address`*`/<0–31>`	Specifies the IP address and subnet mask length for the logical interface.
`logical-name` *`new_log_if_name`*	Specifies a new logical name for the interface or shows the current logical name. If a logical interface is part of an IPSO cluster, do not change its logical name.
`enable | disable`	Enables or disables the logical interface.
`MTU <1500-16,000>`	Specifies the maximum transfer unit for the interface. The value must be an integer. **Default:** `1500`

Transparent Mode

Use transparent mode to allow your IPSO appliance to behave like a layer 2 device such as a bridge. Benefits of this type of network configuration include being able to maintain your current local area network configuration or maintain your existing IP address with your ISP.

Using transparent mode, you configure Ethernet interfaces (including aggregated interfaces) on Nokia platforms to behave like ports on a bridge. The interfaces then forward traffic using layer 2 addressing. You can configure some interfaces to use transparent mode while other interfaces on the same platform are configured normally. Traffic between transparent mode interfaces is inspected at layer 2 while traffic between normal interfaces, or between transparent and normal interfaces, is inspected at layer 3.

Note
Transparent mode does not provide complete bridging functionality such as loop detection or spanning tree.

Configuring Transparent Mode

Use the following commands to create a transparent mode groups and add an interface to a transparent mode group.

```
add xmode
        id <1-2147483647>
        interface logical_if_name
        filter encap <DIX | LLC | SNAP> proto hex_value action
           <forward | discard>
```

Use the following commands to delete a transparent mode group and delete an interface from a transparent mode group.

```
delete xmode id <1-2147483647>
        interface logical_if_name
        filter encap <DIX | LLC | SNAP> proto hex_value action
          <forward | discard>
```

Use the following commands to configure a transparent mode group.

```
set xmode id <1-2147483647>
        state <on | off>
        vrrp_enabled <on | off>
        cross-connect-enabled <on | off>
```

Use the following commands to view transparent mode configurations.

```
show
        xmode id <1-2147483647> cross-connect-enabled
        xmode id <1-2147483647> info
        xmode id <1-2147483647> interfaces
        xmode id <1-2147483647> filters
        xmode id <1-2147483647> stat
        xmode id <1-2147483647> state
        xmode id <1-2147483647> vrrp_enabled
        xmodes
```

Arguments

`id <1-2147483647>`	Specifies an interger associated with a transparent mode group. When you use the argument with the `add xmode` command, you create a transparent mode group.
`interface` *`logical_if_name`*	Specifies the name of the logical interface, for example, eth-s1p1c0.
`filter encap <DIX | LLC | SNAP>`	Specifies the Ethernet frame encapsulation for the filter you are creating or deleting.

`proto` *`hex_value`*	Specifies the hexadecimal value of the protocol that is forwarded or dropped by the filter. Do not include "`0x`" before the hexadecimal value.
`action <forward | discard>`	Specifies whether a filter should forward or discard the specified traffic.
`state <`*`on`* `|` `off>`	Enables or disables a transparent mode group. **Default:** off
`vrrp_enabled <`*`on`* `|` `off>`	Enables or disables VRRP for a transparent mode group. **Default:** off
`cross-connect-enabled <on | off>`	Specifies whether traffic for protocols other than IP and ARP should be forwarded by default: • `on`: Traffic other than IP and ARP should be forwarded (unless blocked by a filter). • `off`: Traffic other than IP and ARP should be discarded (unless allowed by a filter).
`cross-connect-enabled`	Shows whether traffic for protocols other than IP and ARP is being forwarded by default.
`info`	Shows the configuration of the specified transparent mode group.
`interfaces`	Shows the interfaces associated with the spacified transparent mode group.
`stat`	Shows the statistics associated with the specified transparent mode group.

`state`	Shows the state of the specified transparent mode group—0 for disabled and 1 for enabled.
`vrrp_enabled`	Show whether VRRP is enabled or disabled on the specified transparent mode group—0 disabled and 1 for enabled.
`xmodes`	Shows the configuration of all transparent mode groups on the platform.

Link Aggregation

You can aggregate (combine) Ethernet ports so that they function as one logical port with higher bandwidth. For example, if you aggregate two 10/100 mbps ports, they function as a single port with a theoretical bandwidth of 200 mbps. Another benefit of link aggregation is redundancy—if one of the physical links in an aggregated group fails, the other physical links remain active and the logical link continues to function.

You can specify a minimum number of ports that must be active for the logical interface to remain active. If the number of active ports is less than this number, the logical interface is deactivated. This option is particularly useful in VRRP configurations. For example, you might have a VRRP pair in which both the master and backup systems use two aggregated Gigabit Ethernet ports as their external connection. If one of the Gigabit Ethernet ports in the master fails, you probably would prefer that the backup system becomes the master so that there is no loss of bandwidth in the external connection. In this case, you would set the minimum number of active ports to be two.

To configure link aggregation, you create an aggregation group and then add interfaces to it. When you add an interface to an aggregation group, its configuration information is deleted. Be careful not to aggregate the interface that you are using for your CLI connection because doing so breaks your connection to the appliance.

When interfaces participate in an aggregation group, you cannot configure them individually—you configure the group instead, using the appropriate interface

commands. When you use interface commands, use the format `aexx` for the physical interface and the format `aexxc0` for the logical interface. For example, the physical name of a group with the ID 10 is `ae10` and its logical name is `ae10c0`.

You must configure an aggregation group with an IP address and so on.You cannot configure an aggregation group with logical information until you have added an interface to the group.

Configuring Link Aggregation

Use the following commands to create, configure, delete, and view link aggregation information.

```
add linkaggregation
        group <1—1024>
               port phys_if_name [type primary]

set linkaggregation group <1—1024>
        lacp_mode <active | passive | off>
        lacp_timer <short | long>
        min_ports number_of_ports
        port_priority <1-65535>
        system_priority <1-65535>
        txpolicy <L2 | L3 | L4 | round-robin> <enable | disable>

delete linkaggregation
        group <1—1024>
               port phys_if_name

show
        linkaggregation
               groups
               group <1—1024>
                      lacp_mode
                      lacp_timer
                      min_ports
                      port_priority
                      system_priority
                      txpolicy
```

Arguments

`group <1—1024>`	Creates or specifies an aggregation group with the specified ID.
`port` *`phys_if_name`*	Specifies a physical interface to add to or delete from an aggregation group. You must delete all the ports from a group before you can delete the group itself.
`type primary`	Specifies that this is the first port added to the group. When deleting ports, you must delete this port last.
`lacp_mode <active | passive | off>`	Specify `active` to enable dynamic link aggregation and configure the interfaces in the aggregation group to send LACP control traffic repeatedly. Nokia recommends that you use this setting. Specify `passive` to enable dynamic link aggregation and configure the interfaces in the aggregation group to send LACP control traffic only in the following circumstances: • An interface needs to provide information about itself to the other end of the link (as happens when a Nokia interface becomes active and receives a packet). • When a configuration setting changes on the Nokia interface. Specify `off` to disable dynamic link aggregation.

`lacp_timer <short | long>`	Specifies how long IPSO should wait for LACP control packets to determine whether to drop inactive interfaces from a link aggregation group: • `short`: IPSO expects to receive an LACP control packet every 30 seconds on each physical interface in the link aggregation group. If it does not receive a control packet on an interface within 30 seconds, it drops the interface from the group. • `long`: IPSO expects to receive an LACP control packet every 90 seconds on each physical interface in the link aggregation group. If it does not receive a control packet on an interface within 90 seconds, it drops the interface from the group. **Default:** `long`
`linkaggregation`	Shows how many link aggregation groups are configured.
`min_ports` *`number_of_ports`*	Specifies the minimum number of ports in the group that must be active for the logical interface to remain active. **Default:** 1
`port_priority <1-65535>`	Specifies a port priority value. This value identifies all the physical interfaces in the link aggregation group. This value is only an identifier—it does not provide any prioritization of any kind.

`system_priority <1-65535>`	Specifies a system priority value. This value identifies the IPSO system to the device at the other end of the link (which might be connected to other devices on which LACP is enabled). This value is only an identifier—it does not provide any prioritization of any kind.

`txpolicy <L2 | L3 | L4 | round-robin> <enable | disable>`	Specifies a method for distributing outgoing traffic between the aggregated interfaces. • Round Robin: IPSO distributes the outgoing traffic across all the physical interfaces equally. • L2: IPSO distributes the outgoing traffic across the physical interfaces using hash values based on the destination MAC addresses of packets. This is not a suitable choice if the other end of the link is another router (because all packets will have the same destination MAC address). • L3: IPSO distributes the outgoing traffic across the physical interfaces using hash values based on the destination IP addresses of packets. This is not a suitable choice if all packets have the same destination IP address (as might be the case if NAT is used at the other end of the link). • L4: IPSO distributes the outgoing traffic across the physical interfaces using hash values based on the destination TCP or UDP port numbers. **Default:** round-robin **Note** The device at the other end of the link does not need to use the same method for distributing traffic between its aggregated ports.

Link Redundancy

You can configure redundant Ethernet interfaces for resiliency purposes. For example, if you create a link redundancy group that includes two physical interfaces and the active interface fails, the second interface takes over and there is no interruption in service. You might want to use this feature if your IPSO platform is connected to a switch that does not support link aggregation.

There are significant differences between link redundancy (Ethernet bonding) and link aggregation:

- There is no load balancing within a link redundancy group—only one of the interfaces in a group is active at a given time.
- The interfaces in a link redundancy group do not need to be configured identically. For example, they can have different speeds and duplicity settings.
- You can include a link aggregation group within a redundancy group, but a redundancy group cannot be part of an aggregation group.

You can combine interfaces from different network interface cards in a single link redundancy group, and you can create as many as eight link redundancy groups per system. Each group can include as many as eight interfaces. (If you include a link aggregation group, it counts as one redundancy inerface regardless of how many physical ports are in the aggregation group.) An interface can participate in only one redundancy group.

On IP2250 and IP2255 platforms, link redundancy is subject to the same constraints as link aggregation:

- Do not include interfaces on different ADP I/O cards in the same link redundancy group.
- Do not combine any of the built-in Ethernet management interfaces with interfaces on an ADP I/O card to form a redundancy group.
- You can combine management interfaces to create a redundancy group.

When you create a link redundancy group, you must designate a primary interface. This is the default active interface—if the primary interface fails and later returns to service it becomes the active interface again. For this reason you should configure the fastest interface in the group to be the primary interface.

All the interfaces in a link redundancy group must connect to the same device at the other end of the link. You cannot configure a single redundancy group across multiple switches.

Configuring Link Redundancy

Use the following commands to create, configure, and delete a link redundancy group.

```
add linkredundancy
        group <1–1024>
               port phys_if_name

delete linkredundancy
        group <1–1024>
               port phys_if_name
```

Arguments

`group <1–1024>`	Creates or specifies a redundancy group with the specified ID.
`port` *`phys_if_name`*	Specifies a physical interface to add to or delete from a redundancy group. You must delete all the ports from a group before you can delete the group itself.

Use the following commands to view link redundancy settings.

```
show
        linkredundancy
               groups
               group <1—1024>
```

Arguments

`linkredundancy`	Shows how many link redundancy groups are configured.
`groups`	Shows the configuration information of all the link redundancy groups.
`group <1—1024>`	Shows the configuration information of the specified link redundancy group.

Point-to-Point Over Ethernet

Use the commands explained in this section to configure Point-to-Point Over Ethernet (PPPoE).

Configuring Profiles

Use the following commands to add a profile without authentication.

```
add pppoe profile name profile_name interface phys_if_name mode <connect-
   on-demand | keep-alive> noauth
        timeout <30-259200; 300, 60>
        peername name
        description name
        mss mss_value
        mtu <136-1492>
```

Use the following commands to add a profile with authentication.

```
add pppoe profile name profile_name interface phys_if_name mode mode_name
   username name password password
         authtype PAP | CHAP CASE
         timeout <30-259200; 300, 60><
         peername name
         description name
         mss mss_value
         mtu <136-1492>
```

Use the following commands to modify a profile without authentication.

```
set pppoe profile name profile_name interface phys_if_name mode mode_name
   noauth
         timeout time_in_seconds
         peername name
         description name
         mtu mtu_value
```

Use the following commands to modify a profile with authentication.

```
set pppoe profile name profile_name interface phys_if_name mode mode_name
   username name password password
         authtype PAP | CHAP CASE
         timeout time_in_seconds
         peername name
         description name
         mtu mtu_value
```

Use the following command to delete a profile.

```
delete pppoe profile name profile_name
```

NOTE: You cannot delete a pppoe profile if it is associated with a logical interface. You must first delete the pppoe logical interface. See "Configuring PPPoE Logical Interface" on page 74.

Use the following command to view profiles.

```
show pppoe profile
        all
        name profile_name
```

Configuring PPPoE Logical Interface

Use the following commands to add or configure a logical pppoe interface in dynamic mode.

```
add interface pppoe0 mode dynamic profile-name profile_name
        interface-name log_if_name
        enable off | on

set interface pppoe0 mode dynamic profile-name profile_name
        interface-name log_if_name
        enable off | on
```

Use the following commands to add or configure a logical pppoe interface in static mode.

```
add interface pppoe0 mode static local-ipaddress ip_address remote-
   ipaddress ip_address profile-name profile_name
        interface-name log_if_name
        enable off | on

set interface pppoe0 mode static local-ipaddress ip_address remote-
   ipaddress ip_address profile-name profile_name
        interface-name log_if_name
        enable off | on
```

Use the following commands to add or configure a logical pppoe interface in unnumbered mode.

```
add interface pppoe0 mode unnumbered logical-interface log_if_name
   profile-name profile_name
        interface-name log_if_name
        enable off | on
```

```
set interface pppoe0 mode unnumbered logical-interface log_if_name
   profile-name profile_name
        interface-name log_if_name
        enable off | on
```

Use the following command to delete the pppoe logical interface.

delete interface *log_if_name*

Use the following command to modify the pppoe logical interface.

```
set interface log_if_name
        admin-status enable | disable
        link_trap on | off
```

Configuring PPPoE Physical Interface

Use the following command to modify the pppoe physical interface.

set interface pppoe0

```
        admin-status enable | disable
        link_trap on | off
```

Arguments

`profile name` *profile_name*	The name used to identify the profile with the associated logical interface. The profile name may be 1 to 31 characters long.
`interface` *phys_if_name*	Specifies the physical ethernet interface.
`mode <connect-on-demand | keep-alive>`	Specifies the availble connection modes. • connect-on-demand: The interface comes up when IP traffic is generated. • keep-alive: The interface is always up.

noauth	Specifies no authentication will be used.
authtype <PAP | CASE>	If you specify an authentication type, you must also specify the user name and password.
username *user_name*	Specifies the user name when using authentication. The user name may be 1 to 63 characters long
password *pass_word*	Specifies the password the user must log in with. The password may be 1 to 31 characters long.
timeout <30-259200; 300>	If the mode is connect-on-demand, the specified timeout indicates idle timeout. If the mode is keep-alive, the specified timeout indicates connection check period. If no value is specified, the system will use 300 seconds for idle timeout and a value of 300 seconds for connection check period.
peername *name*	Specifies the PPPoE server from which the system will accept connections. When this is not specified, the system will accept connections from any PPPoE server. The name can be 1 to 63 characters long.
description *name*	Specifies an identity that the user may use to remember the profile.
mss *mss_value*	Specifies the size in bytes of the maximum segment size.
mtu <136-1492>	Specifies the size in bytes of the maximum transmission unit.

`mode <unnumbered | dynamic | static>`	Specifies how the logical interface will be assigned an IP address. unnumbered: The interface does not have its own unique IP address. Instead, another interface address is used. NOTE: When using the unnumbered command, the Ethernet interface used in the logical-interface variable must have an IP address associated with it. dynamic: The interface is not configured with any IP address because the address is assigned by the peer during session establishment. static: The local and remote IP addresses must be configured. Use dotted-quad format, for example, 10.0.93.1.
`local-ipaddress` *`ip_address`*	Used with mode static command to specify the local interface address.
`remote-ipaddress` *`ip_address`*	Used with mode static command to specify the remote interfacde address.
`enable <off | on>`	Enables or disables the logical interface.
`admin-status disable| enable`	Enable or disable the logical interface.
`link_trap on | off`	Enable or disable the link-trap for the logical interface.

FDDI Interfaces

Use the commands explained in this section to configure physical and logical FDDI interfaces.

Physical FDDI Interfaces

Use the following commands to configure and view the settings for physical FDDI interfaces.

```
set interface phys_if_name
        active <on | off>
        duplex <half | full>


show interface phys_if_name
        all
        duplex
        status
        statistics
```

Arguments

`active <on | off>`	Specifies whether the interface is active or inactive.
`all`	Shows a variety of information about the interface (not including traffic and error statistics).
`duplex <half | full>`	Specifies the duplex mode in which this FDDI interface will operate. It must be the same as that of other systems in the FDDI ring.
`status`	Shows whether the interface is active or inactive.
`statistics`	Shows traffic and error information for the interface.

Logical FDDI Interfaces

Use the following command to change the IP address of a logical FDDI interface.

```
add interface log_if_name address ip_address/mask <8-31>
```

Use the following command to enable or disable a logical FDDI interface.

```
set interface default_log_if_name
        enable | disable
        logical-name log_if_name
```

Use the following command to view information about a logical FDDI interface.

```
show interface default_log_if_name
        address
        status
        statistics
        logical-name
        all
```

Arguments

ip_address/mask <8-31>	Specifies the IP address and subnet mask length for the logical interface.
default_log_if_name	Specifies the default logical name of the interface (such as `fddi-s1p1c0`).
`enable | disable`	Enables or disables the logical interface.
`address`	Shows the IP address assigned to the logical interface.
`status`	Shows whether the interface is active or inactive.

`statistics`	Shows traffic and error information for the interface.
`logical-name`	Specifies a new logical name for the interface or shows the current logical name.
`all`	Shows a variety of information about the interface (not including traffic and error statistics).

ISDN Interfaces

Use the commands explained in this section to configure physical and logical ISDN interfaces.

Physical ISDN Interfaces

The commands explained in this section let you configure and view the settings for physical ISDN interfaces.

```
set interface phys_if_name
        status <on | off>
        switch-type etsi
        line-topology [point-to-point | multi]
        tei-option [automatic | manual]
        tei-assignment [first-call | power-up]
        tei <0-63>
        local-number number
        local-sub-number number
        logging [warn | info | error]
        disconnect-channel <1 | 2>
```

```
show interface phys_if_name
        all
        status
        switch-type
        line-topology
        tei-option
        tei-assignment
        tei
        local-number
        local-sub-number
        logging
        disconnect-channel
        chan-state
        chan-link
        chan-call-info
        chan-last-call-info
```

Arguments

`all`	Shows all the current configuration information.
`status <on | off>`	Turns the physical interface on or off. You must turn the interface off before making any configuration changes (except logging changes).
`switch-type etsi`	Specifies that the service provider's switch is compliant with ETSI (European Telecommunications Standards Institute) standards. **Default:** etsi
`line-topology [point-to-point | multi]`	Specifies the topology of the ISDN connection. **Default:** multi

`tei-option [automatic | manual]`	Specifies whether the terminal endpoint identifier is assigned automatically or manually. **Default:** automatic
`tei-assignment [first-call | power-up]`	Defines when terminal endpoint identifier negotiation occurs. The `first-call` parameter specifies that TEI negotiation occurs when the first ISDN call is placed or received. The `power-up` parameter specifies that the TEI is negotiated when the system is powered on. **Default:** first-call
`tei <0-63>`	Specifies the terminal endpoint identifier. This argument is valid only if `tei-option` has been set to `manual`.
`local-number` *`number`*	If multiple devices are connected to a single ISDN line, use this argument to prevent the system from answering calls not intended for it. If you configure a local number, the system will answer only calls that have *`number`* as the called-party number in the setup message. If you configure a local number, the system will attempt to answer all calls.You can also configure a local subnumber that the system will check (in addition to the local number).
`local-sub-number` *`number`*	If you configure a local subnumber, the system will answer only calls that have *`number`* as the called-party subaddress in the setup message.

`logging [warn | info | error]`	Specifies the serverity level of ISDN messages logged. Specifying a given severity causes all messages at least that severe to be sent to the system logging utility. • `error`**:** An unrecoverable error condition occurred. • `warning`**:** An event that may require attention occurred. • `informational`**:** A noteworthy event occurred. **Default:** error
`disconnect-channel <1 | 2>`	Disconnects an active call on B channel 1 or 2.
`chan-state <1–2>`	Shows the state of the specified B channel (whether its active or not).
`chan-link <1–2>`	Shows whether the link of the specified B channel is up or down.
chan-call-info <1–2>	Shows status information about the current call on the specified B channel
chan-last-call-info <1–2>	Shows status information about the last call made or received on the specified B channel.

Logical ISDN Interfaces

Use the following command to create logical ISDN interfaces.

```
add interface phys_if_name encapsulation <ppp | multilink-ppp>
```

Arguments

`encapsulation <ppp | multilink-ppp>`	Specifies the encapsulation protococol.

Use the following commands to add and delete incoming numbers for logical ISDN interfaces.

```
add interface log_if_name incoming-number number

delete interface log_if_name incoming-number number
```

Arguments

`incoming-number` *`number`*	Specifies or deletes the remote numbers that can call into this interface. Incoming calls are checked against the incoming numbers configured for each logical interface. A call will be accepted on the first interface configured to accept incoming calls with an incoming number that matches the calling number in the incoming SETUP packet.If no match is found, the call is rejected. (This functionality requires that the network supports calling line identification.) If an interface is configured to accept incoming calls (set to `incoming` or `both`—see page 87 for more information) and is not configured with any incoming numbers, it will accept any incoming calls. Logical interfaces that are not configured to accept incoming calls or that are already connected will not accept incoming calls.

Use the following commands to configure logical ISDN interfaces and view the corresponding configuration settings.

```
set interface log_if_name
        enable | disable
        description description
        direction <outgoing | incoming | both>
        rate <64kbps | 56kbps>
        idle-time <0–999999>
        minimum-call-time <0–999999>
        remote-number number
        remote-sub-number number
        calling-number number
        calling-sub-number number
        local-name name
        local-password password
        remote-auth-method <pap | chap | none>
        remote-name name
        remote-password password
        bandwidth-util-level <0–100>
        bandwidth-util-period <0–999>
        echo-interval <0–255>
        max-echo-failures <0–255>
        max-mrru <0–99999>
        fragment-size <0–99999>
        address ip_address
        destination ip_address
        unnumbered <yes | no>
        proxy-interface if_name
        connect-channel
        lcp-options <magic-number | no-magic-number | mru | no-mru
           |mrru| no-mrru
           |short-seq-num | no-short-seq-num | endpoint-disc
           |no-endpoint-disc>
```

```
show interface log_if_name
        encapsulation
        status
        description
        direction
        rate
        idle-time
        minimum-call-time
        remote-number
        remote-sub-number
        calling-number
        calling-sub-number
        local-name
        local-password
        remote-auth-method
        remote-name
        remote-password
        bandwidth-util-level
        bandwidth-util-period
        echo-interval
        max-echo-failures
        max-mrru
        fragment-size
        address
        destination
        unnumbered
        proxy-interface
        connect-channel
        lcp-options
        incoming-number number
```

Arguments

`enable` | `disable`	Enables or disables the logical interface. **Default:** disable
`encapsulation`	Specifies or shows whether the logical interface is configured to use PPP or multilink PPP encapsulation.

`status`	Shows whether the logical interface is active or inactive.
`description` *`description`*	Specifies or shows a text string usually used to describe the purpose of the connection. To include spaces in the description, enclose the description in quotation marks.
`direction <outgoing | incoming | both>`	Specifies or shows the direction of ISDN calls supported by the interface. **Default:** outgoing
`rate <64kbps | 56kbps>`	Specifies or shows the connection speed for outgoing calls. **Default:** 64kbps
`idle-time <0-999999>`	Specifies or shows the number of seconds an outgoing call is idle before the call is disconnected. A value of 0 indicates that an outgoing call never times out. **Default:** 120
`minimum-call-time <0-999999>`	Specifies or shows the minimum number of seconds a call must be connected before it can be disconnected by an idle timeout. A value of 0 indicates that a call can be disconnected immediately. If the service provider has a minimum charge for each call, it is recommended that the minimum call time be set to 0. **Default:** 120
`remote-number` *`number`*	Specifies or shows the number to call when establishing an outgoing call.
`remote-sub-number` *`number`*	Specifies or shows the subaddress to use when establishing an outgoing call.

`calling-number` *number*	Specifies or shows the calling number to insert in a setup message. If not specified, no calling number is sent.
`calling-sub-number` *number*	Specifies or shows the calling subaddress to insert in a setup message. If not specified, no subaddress is sent.
`local-name` *name*	Specifies or shows the name used to identify this host to a remote host when the remote host attempts to authenticate this host. This name must match the name for this host that is configured on the remote host. To include spaces in the name, enclose the description in quotation marks.
`local-password` *password*	Specifies or shows the password returned to a remote host for PAP authentication or the secret used to generate the challenge response for CHAP authentication. To include spaces in the password, enclose the description in quotation marks.
`remote-auth-method <pap | chap | none>`	Specifies or shows the method this host uses to authenticate a remote host. • `pap`: password authentication protocol • `chap`: challenge handshake password authentication protocol • `none`: no authentication protocol—therefore, `local-name` and `local-password` do not apply

`remote-name` *`name`*	Specifies or shows the name of the remote host that this logical interface connects to. This name must match the name the remote host uses to identify itself. The name is used with the endpoint discriminator to form a multilink protocol bundle on this host.
`remote-password` *`password`*	Specifies or shows the password returned by the remote host specified with `remote-name` for PAP authentication or the secret used to validate the challenge response for CHAP authentication.
`bandwidth-util-level` `<0-100>`	Specifies or shows the percentage of bandwidth utilization at which bandwidth will be dynamically added or subtracted. When the utilization of the interface exceeds the percentage threshold defined by this value for longer than the time specified with `bandwidth-util-period`, a second channel is added. After a second channel has been added, if utilization falls below this threshold for longer than the time specified with `bandwidth-util-period`, the second channel is removed. This argument is valid only for interfaces that use multilink PPP encapsulation. **Default:** 70

`bandwidth-util-period <0-999>`	Specifies or shows the number of seconds utilization must remain above the threshold set by `bandwidth-util-level` before a second channel will be added. Once a second channel has been added, this value specifies the number of seconds that utilization of the interface must remain below the threshold before the second channel is removed. This argument is valid only for interfaces that use multilink PPP encapsulation. **Default:** 10
`echo-interval <0-255>`	Specifies or shows the number of seconds between echo-request message transmissions (which test for an active remote system). This value must match the echo interval time configured on the system at the remote end of the link or the link state will fluctuate. **Default:** 0
`max-echo-failures <0-255>`	Specifies or shows the maximum number of echo-request messages that can be sent without a reply being received. If no reply has been received after this number of requests have been sent, this system will consider the link to be down. **Default:** 30
`max-mrru <0-99999>`	Specifiesor shows the maximum size of a packet when all fragments have been reassembled (a reconstructed receive unit). This argument is valid only for interfaces that use multilink PPP encapsulation and only if you have enabled the `mrru` LCP option. **Default:** 1500

`fragment-size <0-99999>`	Specifies or shows the maximum size of a packet fragment that may be sent over a multilink protocol link. If set to zero, then packets will not be fragmented. This argument is valid only for interfaces that use multilink PPP encapsulation. **Default:** 0
`address` *`ip_address`*	Specifies or shows the IP address of the interface.You must enter a valid IP address. IPSO does not support dynamically assigning IP addresses for ISDN interfaces. Do not enter 0.0.0.0.
`destination` *`ip_address`*	Specifies or shows the IP address of the remote router's point-to-point interface.
`unnumbered <yes | no>`	Specifies or shows whether the interface is unnumbered. In unnumbered mode the interface does not have its own unique IP address—the address of another interface is used. Specify the other interface with the proxy-interface log_if_name form of this command. **Default:** no
`proxy-interface` *`log_if_name`*	Use when unnumbered mode is enabled to specify the interface from which the address for this interface is taken. `log_if_name` is the logical name of the other (numbered) interface.
`connect-channel`	Use this argument to manually initiate a connection call on the B channel.

`lcp-options <magic-number | no-magic-number | mru | no-mru |mrru| no-mrru |short-seq-num | no-short-seq-num | endpoint-disc | no-endpoint-disc>`	Specifies or shows link control protocol options. • `magic-number/no-magic-number`: Specifies whether or not this system can negotiate and use a magic number to detect looped-back links. • `mru/no-mru`: Specifies whether or not this system can negotiate a maximum receive unit value. • `mrru/no-mrru`: Specifies whether or not this system can negotiate a maximum receive reconstructed unit value (which specifies the size of the reassembled multilink PPP packet). This argument is valid only for interfaces that use multilink PPP encapsulation. If you enable this option, then the maximum MRRU that can be negotiated is the value you set with the `max-mrru` argument. • `short-seq-num/no-short-seq-num`: Specifies whether or not to use a short multilink protocol fragment sequence number (12 bits) to reduce fragment header overhead. This argument is valid only for interfaces that use multilink PPP encapsulation. • `endpoint-disc/no-endpoint-disc`: Specifies whether or not this system can negotiate an endpoint discriminator to identify this host to multilink protocol peers. This argument is valid only for interfaces that use multilink PPP encapsulation.

`incoming-number` *`number`*	Shows the remote numbers that can call into this interface.

Loopback Interfaces

If you do not explicitly assign an OSPF router ID, the system will automatically use the address of one of the installed interfaces as the router ID. If the interface that has that address assigned to it goes down, the system will have to tear down and rebuild its OSPF configuration with a new router ID. To prevent this from happening, you can assign an IP address to a loopback interface (which will not go down). The system will choose this address as the router ID instead of using the address of one of the installed interfaces.

You may also want to assign an IP address to a loopback interface so that you can use the loopback interface as the proxy interface for an unnumbered interface. Again, the benefit of using the loopback interface as a proxy is that it will not go down.

Logical Loopback Interfaces

Use the commands explained in this section to configure logical loopback interfaces.

`add interface` *`log_if_name`* `address` *`ip_address`*

`delete interface` *`log_if_name`* `address` *`ip_address`*

`set interface` *`log_if_name`* `logical-name` *`log_name`*

`show interface` *`log_if_name`* `addresses`

Logical or Physical Loopback Interfaces

Use the commands explained in this section to configure logical or physical loopback interfaces.

```
show interface if_name
        status
        disabled-proto
        enabled-proto
        all
```

Arguments

if_name	You can specify the logical or physical name of the interface.
status	Displays whether the interface is active or not.
disabled-proto	Displays protocols that are not enabled for the interface.
enabled-proto	Displays protocols that are enabled for the interface.
all	Displays all configuration options for the interface.

Modem Interfaces

Use the following commands to enable dialup access to the system through a modem and view the current modem settings.

```
set modem <com2 | com3 | com4>
        country-code <<0-99> | <00-FF>>
        enable | disable
        inactivity-timeout <0-5>
        poll-interval <0-59>
        enable-dialback | disable-dialback
        dialback-number phone_num
        type <5oC1 | 5oC2> [country-code <<0-99> | <00-FF>>]


show modem <com2 | com3 | com4>
        active
        inactivity-timeout
        poll-interval
        dialback
        dialback-number
        country-code
        status
        all
```

Arguments

`com2 | com3 | com4`	Specifies the communications port that the modem is using.
`enable | disable`	Enables or disables the modem using the specified communications port.
`active`	Shows whether the modem on the specified port is active.
`inactivity-timeout <0-5>`	Specifies or shows the number of minutes that a call on the modem can remain inactive (no traffic sent or received) before the call is disconnected. Setting the value to 0 disables the timer, and calls will never be disconnected because of inactivity. **Default:** 0

`poll-interval <0-59>`	Specifies or shows the number of minutes between modem "line status" tests. The system will test whether the modem is present and online once every interval. If a modem is not detected or is offline, an appropriate message appears in syslog. Setting the value to `0` disables the test. **Default:** 0
`enable-dialback | disable-dialback`	If you specify `enable-dialback`, the system drops incoming calls to the modem on this port after the user logs in and then calls the dialback number and connects a login process to the line.
`dialback-number` *phone_num*	Specifies or shows the number that the system will call after it drops an incoming call. You can enter commas to cause the dialing to pause briefly. To increase the length of the pause, enter multiple adjacent commas, as in 650,,555,,1212.
`type <5oC1 | 5oC2>`	Specifies the type of PC Card (PCMCIA) modem that is installed. • `5oC1`: Ositech Five of Clubs I • `5oC2`: Ositech Five of Clubs II
`country-code <<0-99> | <00-FF>>`	Specifies or shows the country code of the number that the system will call after it drops an incoming call. • `0-99`: use when the modem type is `5oC1` • `00-FF`: use uppercase hexadecimal when the modem type is `5oC2`
`status`	Shows whether there is a modem on the specified port.

Serial Interfaces

Use the commands explained in this section to configure physical and logical serial interfaces. The commands for configuring physical interfaces are explained immediately below. To learn about configuring logical interfaces, see "Logical Serial Interfaces" on page 117.

Physical Serial Interfaces

Some commands apply to any physical serial interface and others are applicable only to specific types of interfaces.

Any Physical Interface

Use the following commands to configure and view the settings of any physical serial interface.

```
set interface phys_if_name
        active <on | off>
        encaps <chdlc | ppp | fr>
        keepalive <0–255>
        clocking <external | internal>
        queue-mode <disable|min-latency | max-bw>


show interface phys_if_name
        status
        encaps
        keepalive
        clocking
        speed
        queue-mode
```

Arguments

`active <on | off>`	Specifies whether this interface is on or off.

`status`	Shows whether this interface is on or off.
`encaps <chdlc | ppp | fr>`	Specifies the datalink protocol to run over the device. This setting must match the datalink protocol of the system at the remote end of the point-to-point link. • `chdlc`: Cisco HDLC. Using this option creates a logical interface for this physical interface. • `ppp`: point to point protocol. Using this option creates a logical interface for this physical interface. • `fr`: frame relay. Using this option does *not* create a logical interface for this physical interface. To create a logical interface that uses frame relay encapsulation, you must also specify a datalink connection identifier (DLCI). See page 115 for more information.
`keepalive <0–255>`	Specifies the number of seconds between keepalive protocol message transmissions. These messages periodically test to find out if the remote system is active. This value must match the keepalive value configured on the remote system, or the link state will fluctuate. Setting the keepalive to 0 turns prevents this system from sending keepalive transmissions (that is, setting this to 0 turns off link-failure detection). **Default:** If you are using Cisco HDLC or frame relay encapsulation, the default is `10`. If you are using PPP encapsulation, the default is `5`.

`clocking <external | internal>`	Specifies whether this interface generates the clock signal for the line (internal). You must use internal clocking when connecting to another interface that does not provide a clock signal. **Default:** external
`speed`	Shows the clock speed (in bits per second).
`queue-mode <disable | min-latency | max-bw>`	Specifies whether this interface uses a QoS mechanism. You should enable one of the mechanisms only if you also use IPSO's traffic-management features. • `disable`: do not perform congestion prevention • `min-latency`: minimize latency by using a shorter head-of-line queue • `max-bw`: maximize the available bandwidth by using a longer head-of-line queue **Default:** disable

HSSI, X.21, V.35 Interfaces

Use the following commands to configure and view the settings of a physical HSSI, X.21, or V.35 interface.

```
set interface phys_if_name
        channel-mode <full-duplex | loopback>
        speed <1-45000000 | 1-2048000 | 1-10000000>


show interface phys_if_name
        channel-mode
        speed
```

Arguments

`channel-mode <full-duplex | loopback>`	• `full-duplex`: Specifies bidirectional data transfer. This is the usual operating mode of the device. • `loopback`: The device will loop all received external traffic back to the sender and will loop all data transmitted from this system back to it. This mode should be used only for diagnosing configuration problems. **Default:** full-duplex
`speed <1-45000000 | 1-2048000 | 1-10000000>`	If this interface is set to use internal clocking, you must also use this command to set the internal clock speed (bits per second). This determines the data-transfer rate of the point-to-point link. If the interface can generate only certain clock speeds and rate you configure is not one of these values, the next highest valid speed is used. The ranges of possible values are identified below: • HSSI interfaces: `1-45000000` • V.35 interfaces: `1-2048000` • X.21 interfaces: `1-2048000`

T1 Interfaces

Use the following commands to configure and view the settings of a physical T1 interface.

```
set interface phys_if_name
        channel-mode <normal | llb | clb | rlb | plb>
        line-type <short-haul | long-haul>
        cable-length <0—655>
        transmit-loss <0 | -7.5 | -15 | -22>
        receiver-gain <-30 | -36>
        invert-data | noinvert-data
        timeslot channel(s)
        encoding <ami | b8zs>
        framing <sf | esf>
        channel-speed <64Kbps | 56Kbps>
        jitter-attenuator <off | rx | tx>
        jabuffer-depth <32 | 128>
        fdl-type <ansi | none>
        density-enforcer <on | off>
        speed


show interface phys_if_name
        channel-mode
        line-type
        cable-length
        transmit-loss
        receiver-gain
        invert-data
        timeslot
        encoding
        framing
        channel-speed
        jitter-attenuator
        jabuffer-depth
        fdl-type
        density-enforcer
        speed
```

Arguments

`channel-mode <normal | llb | clb | rlb | plb>`	Nokia T1 interface cards have built in CSU/DSU devices and provide the following channel mode options: • `normal`: Normal data transfer. • `llb` (local loopback): The device will loop all transmitted data back into its receiver. This option should be used only for diagnosing configuration problems. • `clb` (local channel loopback): The device will loop all transmitted data back to the internal receiver of this system. This mode should be used only for diagnosing configuration problems. • `rlb` (remote loopback): The device will loop all received external traffic back to the transmitter without processing the incoming data. This mode should be used only for diagnosing configuration problems. • `plb` (remote payload loopback): The device will loop all received external traffic back to the transmitter internally after processing the incoming data. This mode should be used only for diagnosing configuration problems. **Default:** normal

`line-type <short-haul | long-haul>`	Specifies or shows whether this device is connected to a short-haul (DSX-1) or long-haul (CSU) line. • `short-haul`: Select this option if the cable connected to this interface is less than or equal to 655 feet long. • `long-haul`: Select this option when the interface is connected to repeatered lines more than a total of 655 feet long. **Default:** short-haul
`cable-length <0—655>`	Specifies or shows the length of the cable (in feet) from the interface to the remote end of the link. The line build-out value (which allows signal strength to be properly adjusted) is automatically based on this value. This argument is valid only if the line type is set to `short-haul`. **Default:** 224
`transmit-loss <0 | -7.5 | -15 | -22>`	Specifies or shows the line build-out value (the number of decibels by which the transmit signal is decreased). This argument is valid only if the line type is set to `long-haul`. **Default:** -15
`receiver-gain <-30 | -36>`	Specifies or shows the number of decibels by which the received signal is increased using pulse equalization. This argument is valid only if the line type is set to `long-haul`. **Default:** -30

`invert-data | noinvert-data`	You should enable data inversion when you have chosen any of the following settings: • AMI line encoding • 64Kbps channel speed • pulse-density enforcer enabled Data inversion turns the HDLC 0-bit-stuffing mechanism into a 1-bit-stuffing mechanism, which ensures that the data stream meets the rules required by ANSI T1.403. Data inversion is not required when B8ZS encoding or 56Kbps channel speed is configured. These settings guarantee that ANSI T1.403 pulse-density requirements are maintained. **Default:** noinvert-data
`timeslot` *`channel(s)`*	Use this argument to selectively enable or show the DS0 channels that make up the T1 line, that is, configure fractional T1 lines. All the DS0 channels are enabled by default. If you use this argument to specify channels, those channels remain enabled and all the remaining channels are disabled. You can enable multiple channels by specifying ranges (for example, `2-7`) and/or using commas to separate individual channels (for example, `2,5,7`). The acceptable range of channel values is `1–24`. The following are additional examples of valid values: • `2` (all the other channels are disabled) • `2,7` (all the other channels are disabled) • `2,7,11-18` (all the other channels are disabled)

`encoding <ami | b8zs>`	Specifies the line encoding used by the T1 channel. This setting must match the line encoding of the CSU/DSU at the remote end of the link. • `ami`: Specifies alternate mark inversion encoding. When using this encoding, you can also enable the pulse-density enforcer to ensure that there are enough pulses on the line to maintain line synchronization. • `b8zs`: Specifies bipolar 8-zero substitution. When you use this encoding, the pulse-density enforcer is automatically turned off. **Default:** b8zs
`framing <sf | esf>`	Specifies or shows the framing format, which is used to divide the data stream into DS0 channels and to synchronize with the remote CSU/DSU. This setting must match the frame format used by the CSU/DSU at the remote end of the link. • `sf`**:** Specifies superframe (or D4) framing, an older format that combines as many as 12 T1 frames into a superframe. If you use this format, you cannot use the `fdl-type` argument to specify that performance reports should be sent over the facilities data link. • `esf`: Specifies extended superframe framing, which combines 24 T1 frames into a superframe. It also provides CRC checking on frames and a facilities data link for data monitoring and reporting. **Default:** esf

`channel-speed <64Kbps | 56Kbps>`	Specifies or shows the speed of the DS0 channels in the T1 line. This setting must match the channel speed used by the CSU/DSU at the remote end of the link. • `64Kbps`**:** All 8 bits of the channel are used to carry data. This is the most common configuration. • `56Kbps`: Only 7 of the 8 bits of each channel are used to carry data. This option is used on some older trunk lines. **Default:** 64Kbps
`jitter-attenuator <off | rx | tx>`	Jitter attenuation helps eliminate errors caused by random variations in a clock signal. It is usually used to eliminate fluctuations in the clock signal derived from the received bit stream. • `off`: Jitter attenuation is disabled. • `rx`: Jitter attenuation is enabled on the receiver (input). • `rx`: Jitter attenuation is enabled on the transmitter (output). This setting is appropriate when the built-in CSU (on the Nokia T1 interface card) is providing the clock signal for the link. **Default:** rx

`jabuffer-depth <32 | 128>`	Specifies or shows the jitter attenuator buffer length (in bits). The jitter attenuator buffer length is used to ensure the accuracy of the processed clock signal. This argument is valid only if jitter attenuation is enabled. • `32`: The buffer is 32 bits long. This setting is used when high clock accuracy is expected. • `128`: The buffer is 128 bits long. Use this setting when you expect significant amounts of jitter. **Default:** 32
`fdl-type <ansi | none>`	Specifies or shows whether to send performance reports to the remote CSU/DSU. These performance reports are sent on the facilities data link (a dedicated channel) and do not affect data throughput on the T1 link. This argument is valid only if the interface is using extended superframe (`esf`) framing. • `ansi`: Performance reports conforming to the ANSI T1.403 standard will be sent every second. • `none`: No performance reports will be sent. **Default:** ansi

`density-enforcer <on | off>`	Specifies or shows whether to use the pulse-density enforcer to force the transmit and receive data streams to meet the framing rules specified by the ANSI T1.403 standard. Nokia recommends that you enable the pulse-density enforcer when you use AMI line encoding. The enforcer is automatically disabled when B8ZS encoding is configured. **Default:** off
`speed`	Shows the internal clock speed (in bps). This is the data-transfer rate of the point-to-point link.

E1 Interfaces

Use the following commands to configure and view the settings of a physical E1 interface.

```
set interface phys_if_name
        channel-mode <normal | llb | clb | rlb | plb>
        encoding <ami | hdb3>
        framing <e1 | e1-noframe>
        crc4-framing | no-crc4-framing
        timeslot-16-framing | no-timeslot-16-framing
        timeslot channel(s)
        line-type <short-haul | long-haul>
        invert-data | noinvert-data
```

```
show interface phys_if_name
        channel-mode
        encoding
        framing
        crc4-framing_
        timeslot-16-framing
        timeslot
        line-type
        invert-data
```

Arguments

`channel-mode <normal | llb | clb | rlb | plb>`	Nokia E1 interface cards have built in CSU/DSU devices and provide the following channel mode options: • `normal`: Normal data transfer. • `llb` (local loopback): The device will loop all transmitted data back into its receiver. This option should be used only for diagnosing configuration problems. • `clb` (local channel loopback): The device will loop all transmitted data back to the internal receiver of this system. This mode should be used only for diagnosing configuration problems. • `rlb` (remote loopback): The device will loop all received external traffic back to the transmitter without processing the incoming data. This mode should be used only for diagnosing configuration problems. • `plb` (remote payload loopback): The device will loop all received external traffic back to the transmitter internally after processing the incoming data. This mode should be used only for diagnosing configuration problems.

`encoding <ami | hdb3>`	Specifies or shows the line encoding used by the E1 channel. This setting must match the line encoding of the CSU/DSU at the remote end of the link. • `ami`: Specifies alternate mark inversion encoding. When using this encoding, you can also enable the pulse-density enforcer to ensure that there are enough pulses on the line to maintain line synchronization. • `hdb3`: Specifies high density bipolar (order 3) encoding.
`framing <e1 | e1-noframe>`	Specifies or shows the framing format, which is used to divide the data stream into DS0 channels and to synchronize with the remote CSU/DSU. This setting must match the frame format used by the CSU/DSU at the remote end of the link. • `e1`: Specifies E1 framing. • `e1-noframe`: All timeslots will be used for data. If you use this framing format, you cannot use CRC 4 error checking and cannot configure fractional E1 channels.
`crc4-framing | no-crc4-framing`	Specifies or shows whether to use CRC 4 error checking. This argument is valid only if framing is set to `e1`. **Default:** crc4-framing

`timeslot-16-framing` | `no-timeslot-16-framing`	Specifies or shows whether to use timeslot 16 for signaling or whether it should be available for use as a data channel. • `timeslot-16-framing`: Specifies that timeslot 16 is used for line supervision signaling (for example, to determine if a telephone is on-hook or off-hook) and is thus not available to be used as a data channel. Enabling timeslot 16 framing is generally useful in telephony applications. • `no-timeslot-16-framing`: Specifies that timeslot 16 is not used for signaling purposes (and is thus available to be used as a data channel). This argument is valid only if framing is set to `e1`. **Default:** no-timeslot-16-framing

`timeslot` *`channel(s)`*	Use this argument to selectively enable or show the DS0 channels that make up the E1 line, that is, configure fractional E1 lines. All the DS0 channels are enabled by default. If you use this argument to specify channels, those channels remain enabled and all the remaining channels are disabled. You can enable multiple channels by specifying ranges (for example, `2-7`) and/or using commas to separate individual channels (for example, `2,5,7`). The acceptable range of channel values is `1–31`. The following are additional examples of valid values: • `2` (all the other channels are disabled) • `2,7` (all the other channels are disabled) • `2,7,11-18` (all the other channels are disabled) This argument is valid only if framing is set to `e1`.
`line-type <short-haul | long-haul>`	Specifies or shows whether this device is connected to a short-haul (DSX-1) or long-haul (CSU) line. • `short-haul`: Select this option if the cable connected to this interface is less than or equal to 655 feet long. • `long-haul`: Select this option when the interface is connected to repeatered lines more than a total of 655 feet long. **Default:** long-haul

`invert-data` | `noinvert-data`	You should enable data inversion when you are using AMI line encoding. Data inversion turns the HDLC 0-bit-stuffing mechanism into a 1-bit-stuffing mechanism. >> **Default:** noinvert-data

Frame Relay Encapsulation

If you set an interface to use frame relay encapsulation, use the following commands to configure and view the settings of the physical interface.

```
set interface phys_if_name
        dte | dce
        active-status-monitor <on | off>
        lmi-type <ansi | ccitt | fr-consortium>
        n391 <1—255>
        n392 <1—10>
        n393 <1—10>
        dlci-length <10 | 11 | 13>


add interface phys_if_name
        dlci <16—1007>
        [unit <1-255>]


show interface phys_if_name
        dte | dce
        active-status-monitor
        lmi-type
        n391
        n392
        n393
        dlci-length
```

Arguments

dte | dce	Specifies or shows whether the interface operates in DCE or DTE mode. • DTE is typically appropriate when the interface is connected to a frame relay switch. • DCE mode is typically used if the interface is connected directly to a DTE device. **Default:** dte
active-status-monitor <on | off>	Specifies or shows whether to monitor the connection status (and include the status in LMI status messages). You may want to turn off monitoring for testing purposes or when the interface is connected to a device that does not provide this status information. **Default:** on
lmi-type <ansi | ccitt | fr-consortium>	Specifies or shows the LMI protocol used to communicate with a frame relay switch. • ansi: ANSI T1.617 Annex D specification • ccitt: CCITT Q.933 Annex A specification • fr-consortium: frame relay consortium specification **Default:** ansi
n391 <1—255>	This is the LMI full status polling counter (N391)**.** It specifies the number of partial LMI status requests (keepalives) to send before sending a full status request. **Default:** 6

`n392 <1—10>`	This is the LMI error threshold counter (`n392`), which is usedwith `n393` to monitor the reliability of the link. Use this argument to specify the number of errored events that must occur before the link is considered unreliable. **Default:** 3
`n393 <1—10>`	This is the LMI monitored event counter (`n393`), which is usedwith `n392` to monitor the reliability of the link.Use this argument to specify the number of events (out of which there must be the number of errored events specified by the n392 argument) before the link is considered unreliable. This argument also specifies the the number of monitored events that must occur without errors before the link is considered reliable again. If you use the default values for `n392` and `n393`, the link is considered unreliable if three out of four monitored events produce errors. If this occurs, the link is considered reliable again when four monitored events occur without any errors. **Default:** 4
`dlci-length <10 | 11 | 13>`	Specifies or shows the number of bits in the datalink connection identifier (DLCI) in the frame relay address. This argument is valid only if `lmi-type` is set to `fr-consortium`. **Default:** 10
`dlci <16—1007>`	Creates a logical interface with the specified datalink connection identifier.

`unit <1-255>`	Specifies the channel number in the logical name. For example, specifying `unit 4` when you add a logical interface to ser-s3p1 would result in a logical interface named ser-s3p1c4. If you omit the unit parameter, the the next available channel number is used for the logical interface name.

PPP Encapsulation

If you set an interface to use PPP encapsulation, use the following commands to configure and view the settings of the interface.

```
set interface phys_if_name
        keepalive-failures <1—255>
        magic-number | no-magic-number
        mru | no-mru


show interface phys_if_name
        keepalive-failures
        magic-number
        mru
```

Arguments

`keepalive-failures <1—255>`	Specifies or shows the number of times the remote system must fail to respond to keepalive protocol messages before this system considers the link down. **Default:** 30

magic-number | no-magic-number	Specifies or shows whether this system can negotiate and use a magic number to detect looped-back links. This may need to be disabled if the remote system does not support the option. **Default:** magic-number
mru | no-mru	Specifies or shows whether this system can negotiate a maximum receive unit value. This may need to be disabled if the remote system does not support the option. **Default:** mru

Logical Serial Interfaces

Use the commands explained in this section to configure and view logical serial interfaces.

When you configure a physical serial interface to use PPP or Cisco HDLC encapsulation, a logical serial interface is automatically created for that physical interface. If you configure a physical serial interface to use frame relay encapsulation, a logical interface for it is not created until you also specify a datalink connection identifier (DLCI). (See page 98 for information about how to specify the encapsulation format for a physical serial interface. See page 115 for information about how to specify the DLCI for a physical interface.)

```
set interface log_if_name
        logical-name log_if_name
        address ip_address
        destination ip_address
        unnumbered <yes | no>
        proxy-interface log_if_name
        enable | disable
        mtu <128-65535>
```

```
show interface log_if_name
        all
        logical-name
        address
        destination
        dlci
        unnumbered
        proxy-interface
        mtu
        status
        statistics
```

Arguments

`all`	Shows all the current configuration settings.
`logical-name` *`log_if_name`*	Specifies a new logical name for the interface or shows the current logical name.
`address` *`ip_address`*	Specifies or shows the IP address of the logical interface.
`destination` *`ip_address`*	Specifies or shows the IP address of the remote router's point-to-point interface.
`dlci`	Shows the datalink connection identifier of this interface.This is applicable only if the physical interface uses frame relay encapsulation.
`unnumbered <yes | no>`	Specifies or shows whether the interface is unnumbered. In unnumbered mode the interface does not have its own unique IP address—the address of another interface is used. Specify the other interface with the `proxy-interface log_if_name` form of this command. **Default:** no

`proxy-interface` *`log_if_name`*	Use when unnumbered mode is enabled to specify the interface from which the address for this interface is taken. `log_if_name` is the logical name of the other (numbered) interface.
`mtu <128-65535>`	Valid when physical media is HSSI.
`enable` | `disable`	Enables or disables the logical interface. **Default:** disable
`status`	Displays whether the interface is active or not.
`statistics`	Shows traffic and error information for the interface.

VPP Interfaces

Each GPLC in a IP3000 Series system is divided into two “domains.” The local domain is under the control of the GPLC, and the other is under the control of the CRP. Unexported interfaces on the GPLC are part of the local domain, and exported interfaces are part of the domain controlled by the CRP.

Virtual point-to-point (VPP) interfaces provide connections between these domains. They allow processes running in the local domain of a GPLC to communicate and exchange traffic with the CRP and other GPLCs.

The system automatically creates two VPP interfaces for each GPLC. Interface `vpp0` is in the local domain of the GPLC. Interface `vpp1` is in the CRP’s domain—the system automatically exports `vpp1` interfaces to the CRP. The default logical names for these interfaces are `vpp0c0 and vpp1c0.`

If you want the system to pass traffic and communicate between the two domains, you must configure both logical VPP interfaces to work with each other. You can configure VPP interfaces to be numbered or unnumbered.If the VPP interfaces are unnumbered, you may want to use the loopback interfaces on the CRP and GPLC as the proxy interfaces. This approach is advantageous because the loopback interfaces will not go down. See "Loopback Interfaces" on page 93 for more information.

If both of the VPP interfaces are numbered, both of the following must be true:

- the address of the exported interface must match the destination address of the unexported interface
- the address of the unexported interface must match the destination address of the exported interface

If one of the VPP interfaces is numbered, its destination address must be the address

of the proxy interface for the unnumbered interface.

Create Appropriate Static Routes

To pass traffic between unexported interfaces and the CRP, you must also configure appropriate static routes (on both the GPLC that has the unexported routes and the CRP). If the VPP interfaces are unnumbered, you must use `logical-name` as the gateway type when you create the static routes.

VPP Interface Commands

Use the following commands to configure and view the settings for VPP interfaces.

```
set interface log_if_name
        enable | disable
        address ip_address
        destination ip_address
        unnumbered <yes | no>
        proxy-interface log_if_name
        logical-name log_if_name

show interface log_if_name
        all
        status
        address
        destination
        unnumbered
        proxy-interface
        logical-name
```

Arguments

`all`	Shows all the current configuration settings.
`status`	Shows whether the interface is active.
`enable | disable`	Enables or disables the logical interface.
`address` *`ip_address`*	Specifies or shows the IP address of the logical interface.
`destination` *`ip_address`*	Specifies or shows the IP address of the point-to-point interface at the other end of the link.
`unnumbered <yes | no>`	Specifies or shows whether the interface is unnumbered. In unnumbered mode the interface does not have its own unique IP address—the address of another interface is used. Specify the other interface with the `proxy-interface log_if_name` form of this command.
`proxy-interface` *`log_if_name`*	Use when unnumbered mode is enabled to specify the interface from which the address for this interface is taken. `log_if_name` is the logical name of the other (numbered) interface.
`logical-name` *`log_if_name`*	Specifies a new logical name for the interface or shows the current logical name.

3 System Configuration Commands

This chapter describes the system configuration commands that you can enter from the CLI prompt.

System Configuration Summary

Use the following command to view the configuration summary:

```
show summary
```

Arguments

`show summary`	Displays the configuration of the platform.

Configuring Banner and Login Messages

Use the following commands to configure a banner message, an FTP welcome message, and a “message of the day” (MOTD) that users see when they log in using the command line.

```
set message
        banner <on | off> [msgvalue "message"]
        ftpwelcome <on | off> [msgvalue "message"]
        motd <on | off> [msgvalue "message"]
```

```
delete message
        all
        banner
        ftpwelcome
        motd

show message
        all [status]
        banner [status]
        ftpwelcome [status]
        motd [status]
```

Arguments

`banner`	Specifies the banner message that users see when they connect to the system (before they log in) or when they log out.
`ftpwelcome`	Specifies the message that users see when they log into the system using FTP.
`motd`	Specifies the message that users see when they log into the system using the command line.
`all`	Specifies all the messages.
`[msgvalue "`*`message`*`"]`	Specifies the text of the message.
`<on | off>`	Enables or disables the message.
`[status]`	Displays whether a message is enabled or disabled.

Configuring DHCP

Use the following commands to configure DHCP clients and DHCP servers.

DHCP Service Commands

Use the following commands to select the type of DHCP service.

```
set dhcp service
        server
        client
        relay
        none
```

Use the following command to show the type of service.

```
show dhcp service
```

Use the following command to view all DHCP configurations.

```
show dhcp server all
```

Arguments

`server`	Specifies that the server process will be configured on the appliance.
`client`	Specifies that the client process will be configured on the appliance.
`none`	No DHCP process is specified. None is the default.

Configuring DHCP Clients

Use the following commands to add a DHCP client configuration.

```
add dhcp client interface logical_name
        clientid name
        hostname name
        timeout <0-4294967295, 60>
        retry <0-4294967295, 300>
        leasetime <0-4294967295>
        reboot <0-4294967295, 10>
```

Use the following commands to change a DHCP client configuration.

```
set dhcp client interface logical_name
        clientid name
        hostname name
        timeout <0-4294967295, 60>
        retry <0-4294967295, 300>
        leasetime <0-4294967295>
        reboot <0-4294967295, 10>
        enable
        disable
```

Use the following command to delete DHCP clientt configurations.

```
delete dhcp client interface logical_name
```

Use the following commands to view DHCP client configurations.

```
show dhcp client
        interface logical_name
        interfaces
```

Arguments

`client interface` *`logical_name`*	Associates a logical Ethernet interface for the DHCP client to send and receive DHCP messages and configuration parameters from a DHCP server.
`clientid` *`name`*	Creates a unique identifier that is used in place of the MAC address of the client.

`hostname` *name*	Creates a hostname for the client. If you do not specify a host name, the server will name the client.
`timeout <0-4294967295, 60>`	Specifies a time limit, in seconds, for the client to gain an IP address from the server. The default is 60 seconds.
`retry <0-4294967295, 300>`	Specifies a time, in seconds, to retry contacting a server. The default is 300 seconds.
`leasetime <0-4294967295>`	Specifies the time, in seconds, for which the client requests an IP address. No default.
`reboot <0-4294967295, 10>`	Specifies the time, in seconds, after the client first tries to reacquire an IP address and the time the client tries to discover a new IP address.
`interfaces`	When used with the show command, displays all client DHCP interfaces.
`enable`	Enables the DHCP client process.
`disable`	Disables the DHCP client process.

Configuring DHCP Servers

Use the following commands to configure DHCP servers.

Configuring Subnets

Use the following commands to create subnets.

```
add dhcp server subnet ip_address netmask <1-32>
        router ip_address
        default-lease <0-4294967295, 43200>
        max-lease <0-4294967295, 43200>
        domain name
        dns ip_address
        ntp ip_address
        tftp name | ip_address
        wins ip_address
        ddserver ip_address
        note-type <B-node, P-node, M-node, H-node>
        scope name
        zone name
        swap name | ip_address

add dhcp server subnet ip_address
        pool start ip_address end ip_address
```

Use the following commands to change subnet configurations.

```
set dhcp server subnet ip_address netmask <1-32>
        router ip_address
        default-lease <0-4294967295, 43200>
        max-lease <0-4294967295, 43200>
        domain name
        dns ip_address
        ntp ip_address
        tftp name | ip_address
        wins ip_address
        ddserver ip_address
        note-type <B-node, P-node, M-node, H-node>
        scope name
        zone name
        swap name | ip_address
        enable | disable
```

Use the following commands to view subnet configurations.

```
show dhcp server
        subnets
        subnet ip_address
```

Use the following commands to delete subnets.

```
delete dhcp server
        subnets
        subnet ip_address
```

Arguments

`add dhcp server subnet` *`ip_address`* `netmask <1-32>`	Specifies the subnet where the server will listen for DHCP messages from clients.
`router` *`ip_address`*	Specifies the default router clients will use.
`default-lease <0-4294967295, 43200>`	Specifies the IP address lease time, in seconds, that clients will be given if clients do not request a specific lease time. The default is 43200 seconds.
`max-lease <0-4294967295, 43200>`	Specifies the maximum IP address lease time, in seconds, that clients will be given regardless of client requests. The default is 43200 seconds.
`domain` *`name`*	Specifies the domain name clients will be given, for exampl, client_name.nokia.com.
`dns` *`ip_address`*	Specifies the Domain Name Server (DNS) servers for clients, in order of precedents. Use commas to separate addresses, for example, 195.163.25.3, 195.163.24.1, 195.163.23.5, etc.
`ntp` *`ip_address`*	Specifies the Network Time Protocol (NTP) servers for clients, in order of precedents. Use commas to separate addresses, for example, 195.163.25.3, 195.163.24.1, 195.163.23.5, etc.

`tftp` *name* | *ip_address*	Specifies the Trival File Transfer Protocol (TFTP) servers for clients. Use a dotted-quad address or a valid hostname.
`wins` *ip_address*	When configuring NetBIOS, specifies the Windows Internet Naming Server (WINS) servers for clients, in order of precedents. Use commas to separate addresses, for example, 195.163.25.3, 195.163.24.1, 195.163.23.5, etc.
`ddserver` *ip_address*	When configuring NetBIOS, specifies the Datagram Distribution (DD) servers for clients, in order of precedents. Use commas to separate addresses, for example, 195.163.25.3, 195.163.24.1, 195.163.23.5, etc.
`node-type <B-node, H-node, M-node, P-node>`	When configuring NetBIOS, specifies the nodetype the cleint should disignate itself. • B-node: Only broadcast on the local network for NetBIOS resolution and advertising. • H-node: Unicast to WINS servers. If this fails, broadcast. • M-node: Broadcast on local network, unicast to WINS server. • P-node: Only unicast to WINS server for NetBIOS resolution and advertising.
`scope` *name*	When configuring NetBIOS, specifies the scope for the client.
`zone` *name*	
`swap` *name* | *ip_address*	Specifies the server which provides a swap space for clients. Use a dotted-quad address or valid hostname.
`enable` | `disable`	Enables or disables the subnet for DHCP service.

`add dhcp server subnet` *`ip_address`* `pool start` *`ip_address`* `end` *`ip_address`*	Creates a pool of addresses to be leased to clients. The start and end addresses of the pool must belong to the subnet being configured.
`subnets`	When used with the show command, displays all the DHCP subnets configured on the appliance. When used with the delete command, deletes all the DHCP subnets configured on the appliance.
`subnet` *`ip_address`*	When used with the show command, displays the DHCP subnet specified. When used with the delete command, deletes the DHCP subnet specified.

Configuring Fixed-IP Addresses

Use the following commands to assign an IP address to a spacific host.

```
add dhcp server host name
        clientid name
        mac-address mac_address
        address ip_address
        domain name
        file file_name
        dns ip_address
        ntp ip_address
        smtp name
        tftp name | ip_address
        root file_name
        extension file_name
        time value
        swap ip_address
```

Use the following commands to enable or change fixed-ip address configuration.

```
set dhcp server host name
        enable | disable
        clientid name
        mac-address mac-address
        address ip_address
        domain name
        file file_name
        dns ip_address
        ntp ip_address
        smtp ip_address
        tftp name | ip_address
        root file_name
        extension file_name
        time <-43200 to 43200>
        swap ip_address
```

Use the following commands to delete fix-ip configurations.

```
delete dhcp server
        hosts
        host hostname
```

Use the following commands to view fixed-ip configurations.

```
show dhcp server
        hosts
        host hostname
```

Arguments

`dhcp server host name`	Specifies the host name for the client using the fixed-ip address.
`enable | disable`	Enables or disables the allocation of the fixed-ip address to the specified client.
`clientid name`	Specifies a client name which will be used by the server in place of the MAC address of the client.

`mac-address` *`mac_address`*	Specifies the MAC address of the client. If clientid is configured, the clientid will take precedence.
`address` *`ip_address`*	Specifies the IP address to be assigned to the client.
`domain` *`name`*	Specifies the domain name for the client will be given, for exampl, client_name.nokia.com.
`file` *`file_name`*	Specifies the bootfile name for the client.
`dns` *`ip_address`*	Specifies the Domain Name System (DNS) servers for the client, in order of precedents. Use commas to separate addresses, for example, 195.163.25.3, 195.163.24.1, 195.163.23.5, etc.
`ntp` *`ip_address`*	Specifies the Network Time Protocol (NTP) servers for the client, in order of precedents. Use commas to separate addresses, for example, 195.163.25.3, 195.163.24.1, 195.163.23.5, etc.
`smtp` *`ip_address`*	Specifies the Simple Mail Transfer Protocol (SMTP) servers that are availble to the client. Use commas to separate addresses, for example, 195.163.25.3, 195.163.24.1, 195.163.23.5, etc.
`tftp` *`name`* | *`ip_address`*	Specifies the Trival File Transfer Protocol (TFTP) servers for the client. Use a dotted-quad address or a valid hostname.
`root` *`file_name`*	Specifies the full path name to be used as the root disk partition for the client.
`extension` *`file_name`*	Specifies the full path name of the file that contains addtitional options for the client.

`time <-43200 to 43200>`	Specifies the time zone offset, in seconds, from the coordinated universal time the client should use. A postive offset indicates a location east to zero meridian, and a negative offset indicates a location west to zero meridian.
`wins` *ip_address*	When configuring NetBIOS, specifies the Windows Internet Naming Server (WINS) servers for clients, in order of precedents. Use commas to separate addresses, for example, 195.163.25.3, 195.163.24.1, 195.163.23.5, etc.
`ddserver` *ip_address*	When configuring NetBIOS, specifies the Datagram Distribution (DD) servers for clients, in order of precedents. Use commas to separate addresses, for example, 195.163.25.3, 195.163.24.1, 195.163.23.5, etc.
`node-type <B-node, H-node, M-node, P-node>`	When configuring NetBIOS, specifies the nodetype the cleint should disignate itself. • B-node: Only broadcast on the local network for NetBIOS resolution and advertising. • H-node: Unicast to WINS servers. If this fails, broadcast. • M-node: Broadcast on local network, unicast to WINS server. • P-node: Only unicast to WINS server for NetBIOS resolution and advertising.
`scope` *name*	When configuring NetBIOS, specifies the scope for the client.
`swap` *name* | *ip_address*	Specifies the server which provides a swap space for clients. Use a dotted-quad address or valid hostname.

`hosts`	All hosts with fixed-ip addresses.
`host` *`name`*	Specific host named in the variable.

Configuring Dynamic Domain Name System (DDNS) Service

Use the following commands to create an initial DDNS configuration, and enable or disable the configuration.

```
set dhcp server ddns
        update-style <none | interm>
        ttl <0-255>
        enable | disable
```

Use the following commands to create a DDNS key configuration.

```
add dhcp server ddns key name
        algorithm HMAC-MD5-SIG-ALG.REG.INT | none
        secret value
```

Use the following commands to change a DDNS key configuration.

```
set dhcp server ddns key name
        algorithm HMAC-MD5-SIG-ALG.REG.INT | none
        secret name
```

Use the following commands to delete a DDNS key configuration.

```
delete dhcp server key name
```

Use the following commands to view DDNS key configurations.

```
show dhcp server
        keys
        key name
```

Arguments

`update-style`	Specifies the update style for DDNS.
`ttl <0-255)`	Specifies the time to live value, in seconds, for DNS update messages.
`enable | disable`	Enables or disable DDNS service.
`dhcp server ddns key` *`name`*	Specifies the key name identifier when used with the add command.
`algorithm HMAC-MD5-SIG-ALG.REG.INT | none`	Specifies the algorithm used by the associated key.
`secret` *`value`*	Secret to be matched by DNS server for this key.
`keys`	Shows all keys.
`key` *`name`*	Shows specified key.

Configuring Dynamic Domain Name System (DDNS) Zones

Use the following commands to create a DDNS zone.

```
add dhcp server zone name key name primary ip_address
        secondary ip_address
```

Use the following commands to change DDNS zone configurations.

```
set dhcp server zone name key name primary ip_address
        secondary ip_address
        enable | disable
```

Use the following commands to delete DDNS zones.

```
delete dhcp server
        zones
        zones name
```

Use the following commands to view DDNS key configurations.

```
show dhcp server
        zones
        zone name
```

Arguments

`dhcp server zone name key name primary ip_address secondary ip_address`	specifies zone name, associates a key and the primary DNS server. Optionally you can specify a secondary DNS server.
`enable | disable`	Enable or disables DDNS zones.
`zones`	All configured zones.
`zone name`	Specified zone.

Backup and Restore Files

Use the following commands to configure your system to perform manual or regularly scheduled backups.

Manually Backing Up

These commands configure your system to perform a manual backup. The archives created by a manual backup reside in the var/backup/ directory.

```
set backup manual
        on
        filename name
        homedirs <on | off>
        logfiles <on | off>
        package name <on | off>
```

Arguments

`on`	Specifies to perform a manual backup. By default, the backup file contains all the configuration (/config), cron (/var/cron), etc (/var/etc), and IPsec files (/var/etc/ipsec). Export versions of IPSO do not include IPsec files.
`filename` *`name`*	Specifies the name of the file that includes all the backed up files. You must specify this name to configure a manual backup.
`homedirs <on | off>`	Specifies whether to include all home directories in the backup file. **Default:** off
`logfiles <on | off>`	Specifies whether to include all log files in the backup file. **Default:** off
`gplcfiles <on | off>`	Specifies whether to include all GPLC files in the backup file.
`package` *`name`* `<on | off>`	Specifies whether to include a specific package file in the backup file. Package files are not automatically included in a backup file. Enter the filename for the package you want to include in the backup.

Scheduling Backups

Use the following commands to configure your system to perform regularly scheduled backups. The archives produced by scheduled backups reside in the `/var/backup/sched/` directory and are time-stamped.

Use the following commands to create and manage scheduled backups.

Note
The command set below allows you to add a scheduled backup one-time only. To schedule a new regularly scheduled backup, delete the existing scheduled backup and use the set backup scheduled command set to configure a new regularly scheduled backup.

```
add backup scheduled
        filename name
        dayofmonth <1-31>
        minute <0-59>
        dayofweek <1-7>
        hour <0-23>

set backup scheduled
        on
        filename name
        hour <0-23>
        minute <0-59>
        homedirs <on | off>
        logfiles <on | off>
        package name <on | off>
```

Use the following command to delete a previously configured scheduled backup:

```
delete backup scheduled
        dayofmonth <1-31> dayofweek <1-7>
```

Arguments

on	Specifies to perform a regularly scheduled backup. By default, the backup file contains all the configuration (/config), cron (/var/cron), etc (/var/etc), and IPsec files (/var/etc/ipsec). Export versions of IPSO do not include IPsec files.

`filename` *name*	Specifies the name of the file that includes all the backed up files. You must specify this name to configure a regularly scheduled backup.
`homedirs <on | off>`	Specifies whether to include all home directories in the backup file. **Default:** off
`logfiles <on | off>`	Specifies whether to include all logfiles in the backup file. **Default:** off
`package` *name* `<on | off>`	Specifies whether to include a specific package file in the backup file. Package files are not automatically included in a backup file. Enter the filename for the package you want to include in the backup.
`dayofmonth <1-31>`	Specifies which day of the month to schedule the backup. This option applies only to monthly scheduled backups. Use this argument also to delete a previously scheduled backup.
`dayofweek <1-7>`	Specifies which day of the week to schedule the backup. This option applies only to weekly scheduled backups. Use this argument also to delete a previously scheduled backup.
`hour <0-23>`	Specifies which hour of the day to schedule the backup.
`minute <0-59>`	Specifies which minute of the day to schedule the backup.

Transferring Backup Files to a Remote Server

You can transfer backup files to a remote server manually or in an automated manner. To use an automated approach, configure a scheduled backup using the commands explained in "Scheduling Backups" on page 138 and use the commands described in "Configuring Automated Transfers" to configure the system to send the backup files to the remote system. To transfer backup files to a remote server manually, use the commands explained in "Transferring Backup Files Manually" on page 142.

Configuring Automated Transfers

Use the following commands to transfer your backup files to a remote server automatically. If you enable automated transfers, backup files are transferred to the remote server as soon as they are complete, assuming the server is reachable. If the remote server is not reachable, the system waits until the next backup occurs and tries again. Once they have been successfully transferred, the backup files are deleted from the system that created them.

```
set backup auto-transfer
        ipaddr ip_address
        protocol
               ftp ftp-dir path_name
               tftp
```

Use the following command to disable automatic transfers of backup files:

```
delete backup auto-transfer ipaddr ip_address
```

Arguments

`ipaddr` *`ip_address`*	Specifies or deletes the IP address of the system to which application core dumps should be sent.

`protocol ftp ftp-dir` *`path_name`*	Specifies to use FTP when sending application core dumps and also specifies the path to the location where the files will be stored. If you choose FTP, make sure that your server accepts anonymous FTP logins. You cannot use nonanonymous FTP logins to transfer application core files.
`protocol tftp`	Specifies to use TFTP when sending application core dumps. Because TFTP does not work with TFTP servers running on many Unix-based operating systems, Nokia recommends that you use FTP unless you are sure that your TFTP server accepts writes to files that do not already exist on the server.

Transferring Backup Files Manually

Use the following commands to manually transfer your backup files to a remote server:

```
set backup remote
        ftp-site ip_address
        ftp-dir path_name
        ftp-user name
        manual filename [ftp-passwd password]
        scheduled filename [ftp-passwd password]
```

To run an interactive session enter:

```
set backup remote <manual | scheduled> filename
```

To run a machine to machine (MMI) session, enter:

```
set backup remote <manual | scheduled> filename ftp-passwd password
```

Use the following command to disable transfers to an FTP user, site, or directory:

```
delete backup remote
        ftp-site
        ftp-dir
        ftp-user
```

Arguments

`ftp-site` *`ip_addr`*	Specifies the IP address of the remote server to transfer your backup files to.
`ftp_dir` *`path_name`*	Specifies the path of the remote server's directory on which the backup files are saved.
`ftp-user` *`name`*	Specifies the name of the user account for connecting to the FTP site. There is no default, but if you do not specify a user account name, the anonymous account is used.
`manual` *`filename`* `[ftp-passwd` *`password`*`]`	Specifies the name of the manual backup file you want to transfer to the remote server and the optional password to use when connecting to the FTP site.
`scheduled` *`filename`* `[ftp-passwd` *`password`*`]`	Specifies the name of the scheduled backup file you want to transfer to the remote server and the optional password to use when connecting to the FTP site.

Restore Files from Locally Stored Backup Files

```
set restore
        manual filename
        scheduled filename
```

Arguments

`manual` *`filename`*	Specifies to restore your files to the system from a manual backup that is locally stored. Manual backups are stored in the var/backup/ directory.

`scheduled` *`filename`*	Specifies to restore your files to the system from a scheduled backup that is locally stored. Scheduled backups are stored in the /var/backup/sched/ directory.

Caution
Restoring from a backup file overwrites your existing files.

Caution
Make sure that you have enough disk space available on your Nokia platform before restoring files. If you try to restore files and you do not have enough disk space, you risk damaging the operating system.

Note
The system must be running the same version of the operating system and the same packages as those of the backup file(s) from which you restore file(s).

Restore Files from Backup Files Stored on Remote Server

Use the following commands to restore files from backup files previously stored on a remote server. See "Transferring Backup Files to a Remote Server" on page 141 for more information on how to transfer backed up files to a remote server.

```
set restore remote
        filename name
        ftp-site ip_addr
        ftp-dir path_name
        ftp-user user_name
        ftp-passwd password
```

Arguments

`filename` *`name`*	Specifies to restore your files from the filename stored on the remote server.
`ftp-site` *`ip_addr`*	Specifies the IP address of the remote server on which the backup files are stored.
`ftp-dir` *`path_name`*	Specifies the Unix path to the directory on which the backup files are stored.
`ftp-user` *`user_name`*	Specifies the name of the user account for connecting to the FTP site on which the backup files are stored. If a username is not set, enter *`anonymous`*.
`ftp-passwd` *`password`*	Specifies the password to use when connecting to the FTP site. You must change the password whenever the FTP site, FTP directory, or FTP user are changed.

Caution
Restoring from a backup file overwrites your existing files.

Caution
Make sure that you have enough disk space available on your Nokia platform before restoring files. If you try to restore files and you do not have enough disk space, you risk damaging the operating system.

Note
The system must be running the same version of the operating system and the same packages as those of the backup file(s) from which you restore file(s).

Show Backup Commands

Use the following commands to monitor and troubleshoot your backup and restore configuration.

```
show backup
        auto-transfer
               all
               ftp-dir
               ipaddr
               protocol
        manual
               filename
               homedirs
               logfiles
               package name
               packages
        remote ftp-site
               ftp-dir
               ftp-user
               manual filenames
               scheduled filenames
        scheduled filename
               package name
               packages
               homedirs
               dayofmonth
               dayofweek
               hour
               minute
               status
```

Arguments

`auto-transfer all`	Shows all the auto-transfer settings.
`auto-transfer ftp-dir`	Shows the path name of the directory on the remote server where backed up files are stored.
`auto-transfer ipaddr`	Shows the IP address of the remote server that backed up files are stored on.
`auto-transfer protocol`	Shows the protocol used to transfer files automatically.
`manual filename`	Shows the names of the files that have been manually backed up and are stored in the /var/backup/ directory.
`manual homedirs`	Shows whether the home directories are manually backed up.
`manual logfiles`	Shows whether log files are backed up.
`manual package` *`name`*	Shows whether a specified package is backed up.
`manual packages`	Shows the names of the package files that have been manually backed up and are stored in the /var/backup/ directory
`remote ftp-site`	Shows the IP address of the remote server that backed up files are stored on.
`remote ftp-dir`	Shows the path name of the directory on the remote server where backed up files are stored.
`remote ftp-user`	Shows the name of the user account used to connect to to the remote server where backed up files are stored.
`remote manual filenames`	Shows the names of the files that have been manually backed up and stored on the remote server.

`remote scheduled filenames`	Shows the names of the files that have been backed up through scheduled backups and are stored on the remote server.
`scheduled filename`	Shows the name of the scheduled backup files stored in the /var/backup/sched/ directory.
`scheduled homedirs`	Shows whether home directories are backed up.
`scheduled gplcfiles`	Shows whether GPLC files are scheduled for backup.
`scheduled logfiles`	Shows whether log files are scheduled for backup.
`scheduled package` *`name`*	Shows whether a specified package is scheduled for back up.
`scheduled packages`	Shows the names of the packages backed up through scheduled backups and stored in the /var/backup/sched directory.
`scheduled dayofmonth`	Shows the day of the month on which regular monthly backups are scheduled.
`scheduled dayofweek`	Shows the day of the week on which regular weekly backups are scheduled.
`scheduled hour`	Shows which hour of the day regular backups are scheduled.
`scheduled minute`	Shows which minute of the day regular backups are scheduled.
`scheduled status`	Shows whether regular backups are scheduled and the date and time of scheduled backups.

Schedule Jobs Through Crontab File

Use the following commands to configure your system to schedule regular jobs. The cron daemon executes jobs on the dates and times you specify.

Scheduling Jobs

```
set cron
        job name command name
        job name command name timezone <local | utc> dayofmonth <1-31>
        job name command name timezone <local | utc> dayofweek <0-7>
        job name command name timezone <local | utc> hour <0-23>
        job name command name timezone <local | utc> minute <0-59>
        job name on
        mailto email_addr
```

Adding Jobs

Use the following commands add new regular jobs:

```
add cron job name command name timezone <local | utc>
        dayofmonth <1-31>
               hour <0-23>
               minute <0-59>
        dayofweek <0-7>
               hour <0-23>
               minute <0-59>
        mailto email_addr
```

Deleting Jobs

Use the following commands to delete scheduled regular jobs.

```
delete cron
        job name
        job name dayofmonth <1-31>
        job name dayofweek <0-7>
        mailto email_addr
```

Arguments

`job` *name*	Specifies a name for a job for the cron daemon to execute. Use alphanumeric characters only and do not include spaces. Use the command name argument to associate the job name with a specific Unix command.
`command` *name*	Specifies the name of the command for the cron daemon to execute. The command can be any Unix command. Associate this command name with a job name.
`timezone <local | utc>`	Specifies which time zone to use to set the configured time. Local refers to the time zone configured on your platform. UTC refers to universal time coordinated, which is kept in the "i" laboratory, where i is any laboratory cooperating in the determination of UTC. In the U.S., the official UTC is kept by the U.S. Naval Observatory.
`dayofmonth <1-31>`	Specifies the day of the month for the cron daemon to execute the scheduled job. Use this argument only to schedule monthly jobs.
`dayofweek <`*0*`-7>`	Specifies the day of the week for the cron daemon to execute the scheduled job. Use this argument only to schedule weekly jobs.
`hour <0-23>`	Specifies the hour of the day for the cron daemon to execute the scheduled job.

`minute <0-59>`	Specifies the minute of the day for the cron daemon to execute the scheduled job.
`job` *`name`* `on`	Enables the specified job name
`mailto` *`email_addr`*	Specifies the email address for the system to send mail regarding your scheduled jobs.

Show Cron Commands

Use the following commands to monitor and troubleshoot your job scheduler configuration.

```
show cron
        job name command
        job name dayofmonth
        job name dayofweek
        job name hour
        job name minute
        jobs
        mailto
```

Arguments

`job` *`name`* `command`	Shows the Unix command associated with the specified job name.
`job` *`name`* `dayofmonth`	Shows the day of the month on which the job associated with the specified job name is executed by the cron daemon for a monthly scheduled job.
`job` *`name`* `dayofweek`	Shows the day of the week on which the job associated with the specified job name is executed by the cron deamon for a weekly scheduled job.

`job` *`name`* `hour`	Shows the hour of the day on which the job associated with the specified job name is executed by the cron daemon.
`job` *`name`* `minute`	Shows the minute of the day on which the job associated with the specified job name is executed by the cron daemon.
`jobs`	Shows only the names of jobs are scheduled for the cron daemon to execute.
`mailto`	Shows the email address to which the system sends information regarding scheduled jobs.

System Failure Notification Configuration

Use this group of commands to configure system failure notification.

Note
You must first configure mail relay before you configure system failure notification.

Enabling System Failure Notification

Use the following command to enable system failure notification:

```
set notify
        onfail <on | off>
```

Use the following command to configure a user name or e-mail address for notification of a system failure:

```
add notify onfail
        recipient name
```

Use the following command to delete a user name or e-mail address for notification of a system failure:

```
delete notify onfail
        recipient name
```

Show System Failure Notification

Use the following commands to view the system failure notification configuration:

```
show notify onfail
        all
```

Arguments

`onfail` *name*	Specifies an e-mail address or user name to which to send email when there is a system failure. If no email address is specified, the email will be sent to the email address specified in Mail Relay. If you have purchased a support contract, you are encouraged to set this field to system-failure@iprg.nokia.com. This will allow Nokia Support to track more easily and respond to system failures.

DNS

Setting DNS

Use this group of commands to configure the domain name and domain name servers for your platform:

```
set dns
        domainname name
        primary ip_address
        secondary ip_address
        tertiary ip_address
        alternate <on | off>
```

Show DNS

Use the following commands to view your DNS configurations:

```
show dns
        dns domainname
        dns primary
        dns secondary
        dns tertiary
        alternate
```

Deleting DNS

Use the following commands to delete DNS configurations:

```
delete dns
        domainname
        primary
        secondary
        tertiary
```

Arguments

`domainname <name>`	Specifies the name that is put at the end of all DNS searches if they fail. This name should be your local domain name and should begin with an alphabetic letter and may consist only of alphanumeric characters and hyphens. Domain names that are also numeric IP addresses are not allowed.
`primary <IPv4 address>`	Specifies the IP address of the first server to use when resolving hostnames.This address should be a host running a DNS server.
`secondary<IPv4 address>`	Specifies the IP address of the server to use when resolving hostnames if the primary server does not respond. This address should be a host running a DNS server.
`tertiary <IPv4 address>`	Specifies the IP address of the server to use when resolving hostnames if the primary and secondary servers do not respond. This address should be a host running a DNS server.
`alternate <on | off>`	Specifies whether to force IPSO to query the secondary or tertiary DNS server if a lookup fails because the queried domain does not exist (if the response is a “non-existent domain” error.)

Static Host Address Assignment Configuration

Use this group of commands to configure static host names for particular IP addresses.

Adding New Host Names

Use the following command to add a new static host name and associate it with an IP address:

```
add host
        name name ipv4 ip_address
```

Modifying Host Names

Use the following command to change an existing static host name and IP address:

```
set host name name ipv4 ip_address
```

Deleting Host Names

Use the following command to delete a static host name and IP address:

```
delete host name name
```

Showing Host Names

Use the following commands to view static host names and IP addresses:

```
show host
        names
        name name ipv4
```

Arguments

`name` *`name`* `ipv4` *`ip_address`*	Specifies the name of a new or existing static host and the associated IP address. The name must be alphanumeric characters, dashes ('-'), and periods ('.'). Periods must be followed by a letter or a digit. The name may not end in a dash or a period. The IPv4 address to be associated with a static hostname must be in a dot-delimited format with the following range: [0-255].[0-255].[0-255].[0-255].
`names`	Displays all the static host names and addresses on the platform.

Host Name Configuration

Use this group of commands to configure the host name of your platform.

Use the following commands to view or change your platforms host name:

`show hostname`

`set hostname` *`name`*

Arguments

`hostname` *`name`*	When you use the argument with the show command and without the variable, the command shows the current host name of your platform. When you use the argument with the set command, it changes the name of your platform to the name indicated in the variable.

Managing IPSO Images

Use this group of commands to view, select, download and test IPSO images.

Note
Flash-based systems can store a maximum of two IPSO images plus Check Point packages.

Show IPSO Images

Use the following commands to view IPSO images stored on your platform:

```
show
        images
        image current
        image testboot
```

Deleting IPSO Images

Use the following command to delete an IPSO image from your platform:

```
delete image <name | last-download>
```

Test Boot, Reboot, and Halt IPSO Images

Use the following commands to test boot an IPSO image:

```
testboot
        image <name | last-download>
              save
        keep
        cancel
```

Use the following command to reboot your platform with a specified IPSO image:

```
reboot
        image <name | last-download>
               save
```

Use the following command to halt the platform with the option to specify an image to use on the next boot:

```
halt
        image <name | last-download>
               save
```

Downloading IPSO Images

Use the following command to download an IPSO image to your platform:

> **Note**
> The download command maintains all currently active packages after a reboot. Use the disable-packages argument as specified below to disable installed packages after a reboot.

```
download image
        url name <disable-packages>
               http-realm name user name passwd name <disable-packages>
```

Arguments

`images`	The IPSO images on your platform.
`image current`	The currently running IPSO image on your platform
`image testboot`	Displays the image being executed while the in test boot mode.

`delete image <name | last-download>`	Deletes the specified image name. The last-download argument deletes the image last downloaded.
`testboot image <name | last-download>`	Reboots your system for a test of an IPSO image. The testboot command works on all platforms except for the IP400 series. If you do not execute a testboot keep command within five minutes of a test boot, the platform will reboot with the previous image. The last-download argument specifies to use the image most recently downloaded.
`reboot image <name | last-download>`	Reboots your system with the specified IPSO image. The last-download argument specifies to use the image most recently downloaded.
`halt image <name | last-download>`	Halts the system and specifies the image to use the next time the system is started.
`save`	Saves any unsaved configuration changes prior to booting.
`keep`	Accepts the IPSO image being tested as the default image. You do not have to reboot.
`cancel`	Immediately cancels the test boot and reboots your platform with the previous image.
`url name`	Specifies an http or ftp url in dot delimited format. If you want the path to be absolute to your home directory, you must start the directory name from which you want to download with %2F, for example, ftp://10.1.1.1/%2Ftmp/ipso.tgz.
`http-realm name`	Specifies the HTTP realm to which authentication is needed. The name must be printable characters, for example, download.

`user` *`name`*	Specifies a login name if one is required to access the ftp or http server. The format must be printable characters.
`disable-packages`	Specifies to deactivate installed packages after the next reboot. The default is for installed packages to remain active after a reboot.
`passwd` *`name`*	Specifies a password if one is required to access the ftp or http server. The format can be any characters.

Managing Configuration Sets

Use this group of commands to create and manage configuration database files.

Configuration Set Commands

Use the following commands to view the current configuration database files and the current state of the active configuration:

```
show
        cfgfiles
        config-state
```

Use the following command to copy the configuration of the running state to the active configuration database file:

```
copy running-config startup-config
```

Use the following command to select a configuration database to become the current running state:

```
load cfgfile name
```

Use the following commands to save or create configuration database files:

```
save
        config
        cfgfile name
        factory-cfg name
```

Use the following command to delete a configuration database file:

```
delete cfgfile name
```

Arguments

`cfgfiles`	Displays all the configuration database files on your platfrom. In the following example, the file titled *initial* is the active configuration file indicated by the word *active* in the left hand column: cfgfile active.prev active initial cfgfile initial_3.6v13 cfgfile initial_3.6v15
`config-state`	Specifies the current state of the active configuration, which can be either unsaved or saved: • unsaved—a change has been made to the configuration which has not been written to the configuration database file. • saved—the configuration of the system matches the current configuration file.

`cfgfile` *name*	When you use this argument with the load command, you will apply the configuration of the database file in the variable to the currently running system. The command produces a warning message that indicates that unsaved configuration changes are lost and a telnet connection may be lost. The name variable can be “default,” in which case any unsaved configuration changes are lost and /config/active is loaded. When you use this argument with the save command, you will save the current state of the system to a file named in the variable. The name variable can be “default,” in which case any unsaved configuration changes are saved to /config/active. When you use this argument with the delete command, the configuration database file you named in the variable will be deleted. You cannot delete the active configuration file.
`config`	Saves the current running state to the current configuration database.
`factory-cfg` *name*	Creates a new factory default configuration, which is saved in a file named in the variable. The factory default configuration will not bring up any interfaces that you have configured in the new configuration database.
`running-config` `startup-config`	Applies the current running configuration to the active configuration database. This is a Cisco-like command.
name	You may use alphanumeric, dash, dot and underscore characters with no spaces for name variable.

Mail Relay Configuration

Use this group of commands to configure mail relay service.

Mail Relay Commands

Use the following commands to configure the location of a mail hub to which locally originated mail will be relayed via SMTP and the remote user to whom the mail is sent.

```
set mail-relay
        server name
        username name
```

Use the following commands to view the mail server and user configurations:

```
show mail-relay
        server
        username
```

Arguments

`server` *name*	Specifies the IP address or hostname of a mail server that will relay outgoing mail. You must use a host name or IP address in a dot-delimited format.
`username` *name*	Specifies the username on the mail server to which mail addressed to admin or monitor will be sent, for example, admin@localhost. **Default:** root

System Logging Configuration

Use the commands described in this section to configure system logging. Systems with and without hard disks have different logging commands and functionality. See "Logging Commands (Flash-Based Systems)" on page 168 for information about the commands for flash-based systems.

Logging Commands (Systems with Disks)

Use the following commands to accept system log messages from other platforms and to specify that your platform logs configuration changes made by authorized users:

```
set syslog
        accept-remote-log <yes | no>
        auditlog <disable | transient | permanent>
        auditlog-presentation text <enable | disable>
        filename name
        voyager-auditlog <on | off>
```

Use the following commands to specify a remote host to receive system log messages:

```
add
        syslog log-remote-address ip_address
               level <emerg | alert | crit | err | warning | notice
               info | debug | all>
        logging ip_address
```

Use the following command to specify the severity level of system log messages sent to a remote host:

```
set logging trap <0-7>
```

Use the following commands to delete a remote host to receive system log messages:

```
delete
        logging ip_address
        syslog log-remote-address ip_address
               level <emerg | alert | crit | err | warning | notice
               info | debug | all>
```

Use the following commands to view system log configurations:

```
show
        logging
        syslog all
        syslog log-remote-address ip_address
        syslog log-remote-addresses
        syslog auditlog
        auditlog-presentation text
        syslog filename
        syslog voyager-auditlog
```

Arguments

`accept-remote-log <yes | no>`	Specifies whether network system log messages should be accepted from other platforms.If this option is set to 'no', network syslog packets are silently ignored. Otherwise network syslog packets are tagged with the sending machine's hostname and logged as if the messages had been generated locally. **Default:** no
`auditlog <disable | transient | permanent>`	Specifies or shows if the system is logging configuration changes. When you enable the auditlog, you must also specify a destination file with `set syslog` command. • `disable`: Disables audit log. • `transient`: Log only transient changes. • `permanent`: Log transient changes and changes that have been saved and will persist after a reboot. **Default:** disable **Note**: This setting is not saved in the configuration file. You must reset it after rebooting.

`auditlog-presentation text <enable | disable>`	Specifies or shows whether the system displays certain log messages in a text format that is more useful than the default format. **Default:** disable
`filename` *`name`*	Specifies destination log file when you enable auditlog. **Default:** /var/log/messages
`voyager-auditlog <on | off>`	Specifies to set the system to log all Apply and Save actions to the Voyager pages. The log records these actions whether or not the operation succeeded. **Default:** off
`log-remote-address` *`ip_address`*	Specifies the IP address of a remote system to which this system will send system log messages. Be careful not to configure two machines to send logs to each other directly or indirectly. Doing so creates a syslog forwarding loop, which causes syslog messages to be repeated indefinitely on both machines.
`level <emerg | alert | crit | err | warning | notice | info | debug | all>`	When you use the syslog log-remote-address command, specifies an associated severity level for each system log message. A remote system is sent some portion of the locally generated system logging messages. Specifying a given severity means that all messages at least that severe are sent to the associated remote host. **Note**: until you configure at least one severity level for a given remote host, the remote host is not sent any system log messages. If you specify multiple severities, the most general least severe severity always takes precedence.

`logging trap <0-7>`	Specifies the severity level for each system log message sent to a remote host. The severity levels are as follows: 0 emerg 1 alert 2 crit 3 err 4 warning 5 notice 6 info 7 debug **Default:** 6
`logging`	Any command using this argument is a Cisco-like command.

Logging Commands (Flash-Based Systems)

Use the following commands to configure logging on flash-based systems:

```
set syslog
        auditlog <disable | transient | permanent>
        flush-frequency <1-24>
        local-log <on | off>
        network-log <on | off>
        primary-log-server ip_address
        secondary-log-server ip_address
        threshold percent
```

Use the following commands to delete a remote host address so that it no longer receives system log messages:

```
delete syslog
        primary-log-server ip_address
        secondary-log-server ip_address
```

Arguments

`auditlog <disable | transient | permanent>`	Specifies whether the flash-based system logs configuration changes. If you enable the auditlog, the messages are saved to `/var/log/messages`. • `disable`: Disables the audit log. • `transient`: Log only transient changes. These are changes to the active configuration file that have not been saved and will not persist after a reboot. • `permanent`: Log transient changes and changes that have been saved and will persist after a reboot. **Default:** `disable`
`flush-frequency <1-24>`	When the specifed number of hours elapses, log messages are transferred to the remote server and the log buffer is cleared regardless of how many messages are in the buffer. You can use this option in combination with `threshold` for saving messages. **Default:** 4
`local-log <on | off>`	Specifies whether the system saves log files to an installed optional disk (flash memory PC card or hard disk). If you enable local logging, log messages are saved in `/var/log/messages` on the optional disk. The messages are saved to the optional disk according to the setting of the `flush-frequency` option.You can save log files to a remote log server and an optional disk simultaneously. See "Optional Disk Configuration (Flash-Based Systems)" on page 172 for information about configuring an optional disk. **Default:** off

`network-log <on | off>`	Specifies whether system log messages will be sent to a remote log server. **Default:** off
`primary-log-server` *`ip_address`*	Specifies or deletes the IP address of a remote log server to which the flash-based system will send system log messages.
`secondary-log-server` *`ip_address`*	Specifies or deletes the IP address of a remote log server to which the flash-based system will send system log messages if the primary log server is not reachable.
`threshold` *`percent`*	Sets the threshold level for saving log messages to the remote server. Flash-based systems can hold 512 log messages in a specific memory buffer. Use this option to control when the messages are saved to the remote server and the buffer is cleared. For example, assume that the threshold percentage is 50 percent. When there are 256 messages in the buffer, the messages are transferred to the remote server and the buffer is cleared. Setting the option to 0 causes the messages to be transferred immediately and not stored in the buffer. Do not use a percent symbol. **Default:** 0

Use the following commands to view system log configurations:

```
show syslog
        all
        auditlog
        flush-frequency
        local-log
        network-log
        primary-log-server
        secondary-log-server
        threshold
```

Arguments

`all`	Shows all the current configuration settings.
`auditlog`	Shows whether the system logs configuration changes.
`flush-frequency`	Shows the frequency (in hours) at which log messages are saved to the remote server.
`local-log`	Shows whether the system is configured to save log messages to an optional disk (flash memory PC card or optional hard disk).
`network-log`	Shows whether the system is configured to save log messages to a remote log server.
`primary-log-server`	Shows the IP address of the primary remote log server.
`secondary-log-server`	Shows the IP address of the secondary remote log server.
`threshold`	Shows the threshold level for saving log messages to the remote server.

Optional Disk Configuration (Flash-Based Systems)

On flash-based platforms, you can add a hard disk (in some platforms) or flash memory PC card so that you can store the following kinds of files locally:

- **Log files**—local log files are deleted whenever a flash-based platform is rebooted. You can configure an optional disk to locally store log files so that they survive reboot.
- **Package files**—on select platforms, you can configure the optional disk to store application packages. Doing so frees up space in the built-in flash memory.
- **Kernel dump files**—on select platforms, you can configure the optional disk to store kernel core dump files. This allows the platform to store kernel core-dump files much larger than those allowed by the swap space allocated for kernel core dumps in the built-in flash memory.

The above options are mutually exclusive: in other words, you cannot configure an optional disk to store both logs and kernel dump files.

When you select a hard disk or PC card as an optional disk, any existing data on the device is erased. If you remove a PC card that contains log files and want to permanently store the data, insert the card into a PC or other computer and save the data to that system before reinserting the card into a Nokia flash-based platform. For instructions on installing a flash-memory PC card or a hard disk, see your platform installation guide. After you install an optional disk and configure it to store files, you must reboot the system to make it available for use.

Note
Use only PC card flash memory that is supported for your platform. If you attempt to use PC card flash memory that has insufficient capacity, the CLI reports it as being too small and you will be unable to configure as an optional disk.

Note
If you are configuring an optional disk to store logs, you must also configure the system to store logs on the optional disk. See "System Logging Configuration" for more information.

Configuring an Optional Disk

Use the following commands to enable or disable an optional disk and to specify what files should be stored on it:

```
set optional-disk device-id < n >
        type <log | pkg | kernel-dump> on [force]
        off
```

Note
In some cases there might be a long delay before the enabling of an optional disk completes. The CLI prompt will not reappear until the operation is complete.

Use the following command to see whether an optional disk is present and enabled:

```
show optional-disks
```

Arguments

`device-id < n >`	Specifies the device to be configured. Use the `show optional-disks` command to display the device IDs associated with the optional disks you have installed.

`type <log | pkg | kernel-dump>`	Specifies whether the optional disk should contain log files, package files, or kernel core files. Not all options are supported on all platforms. You can store package files on an optional disk only on IP265 systems and can store kernel core files on an optional disk only on IP225x systems.
`on`	Enables a hard disk or a PC card as an optional disk. After enabling an optional disk, you must reboot the system.
`force`	Forces the CLI to configure an unlabeled but supported hard disk. If an optional hard disk is unlabeled, the CLI will report it as possibly unsupported. If your hard disk is supported, use the force keyword, which forces the CLI to label and configure the optional hard disk.
`off`	Disables the storing of files on the optional disk. After disabling an optional disk, you must reboot the system

Core-Dump Server Configuration (Flash-Based Systems)

On flash-based platforms, application core files are stored in memory in the directory / var/tmp. When the file system is 95% filled, flash-based systems delete older core files to make room for newer ones.

Similarly, flash-based platforms store IPSO kernel core files in the internal compact flash memory card and can store a maximum of two at a time. If necessary, the older core file is deleted to make room for a new file.

You can configure flash-based systems to transfer both application and kernel core files to a remote server so that older files are retained. If you do so, the application core files are transferred to the remote server on a predetermined schedule that is not configurable by users. Kernel core dump files are sent to the remote server after the system recovers from the problem that caused the core dump. You can verify that the core file was successfully transferred by checking the log message file for a message similar to the following:

```
[LOG_NOTICE] xfer_crash: Transferred kernel core file to
ftp_server_IP_address
```

This message is not displayed on the console.

After core files are transferred to a remote server, they are deleted from memory.

Note
Certain platforms, such as the IP2250 and IP2255, permit you to store kernel core files on an optional disk. If you have configured an optional disk to store kernel core files, you can still configure the remote core dump server feature, allowing the core file on the optional disk to be transferred to the remote server.

Sending Core Files to a Remote Server

Use the following commands to configure the system to send application and kernel core files to a remote server:

```
set dumpserver
        ipaddr ip_address
        protocol
               ftp ftp-dir path_name
               tftp

delete dumpserver ipaddr
```

```
show dumpserver
        all
        ftp-dir
        ipaddr
        protocol
```

Arguments

`ipaddr` *`ip_address`*	Specifies or deletes the IP address of the system to which core files should be sent.
`protocol ftp ftp-dir` *`path_name`*	Specifies to use FTP when sending core files and also specifies the path to the location where the files will be stored. If you choose FTP, make sure that your server accepts anonymous FTP logins. You cannot use nonanonymous FTP logins to transfer application core files.
`protocol tftp`	Specifies to use TFTP when sending core files. Because TFTP does not work with TFTP servers running on many Unix-based operating systems, Nokia recommends that you use FTP unless you are sure that your TFTP server accepts writes to files that do not already exist on the server.

Date and Time Configuration

Use the following commands to manually configure the date and time on your system:

Setting Date and Time from Server

```
set date
        once-from-ntpserver <ip_address | fully qualified domain name>
        timezone-city value
```

Note
To display a complete list of timezone values, press tab after `timezone-city`.
The default value is Greenwich(GMT)

```
day <1-31>
hour <0-23>
minute <0-59>
second <0-59>
month <1-12>
year 4 digit integer value
```

Setting Date and Time Manually

You can also use the one of the following 2 commands to set the date and time:

```
set clock time month date year
```

```
set clock time date month year
```

Arguments

`once-from ntpserver <ip_address | fully qualified domain name>`	Specifies to set the local time by contacting the NTP server. Enter either the NTP server's IP address or fully qualified domain name.
`timezone-city value`	Specifies a time based on the time zone of a particular place. The default is Greenwich Mean Time (GMT). To display the complete list of values, press tab after `timezone-city`.
`day <1-31>`	Specifies which day of the month to use to set the inital time.
`hour <0-23>`	Specifies which hour of the day to use to set the inital time.

`minute <0-59>`	Specifies which minute of the hour to use to set the initial time.
`second <0-59>`	Specifies which second of the minute to use to set the initial time.
`month <1-12>`	Specifies which month of the year to use to set the initial time
`year` *`4 digit integer value`*	Specifies which year to use to set the initial time. For example, enter *`2002`*. The range is 1970-2037.

The following table explains the arguments for the `set clock` command set.

Arguments

`time`	Specifies the time. Use the following format: 2 digits for the hour:2 digits for the minute:2 digits for the seconds. For example, 15:18:30
`month`	Specifies the month of the year. Enter one of the following: jan; feb; mar; apr; may; jun; jul; aug; sep; oct; nov; dec.
`date`	Specifies the date Enter 1-31.
`year`	Specifies the year. Enter a 4 digit value.

Show Date and Clock Commands

Use the following commands to view your date and time settings:

```
show date

show date timezone-city

show clock
```

Arguments

`date`	Displays the system's configured date and time in the following format: day of the week; month; date time year; timezone. For example: *`Mon Mar 18 22:16:51 2002 GMT`*
`date timezone-city`	Displays the system's configured time only. For example: *`Greenwich (GMT)`*.
`clock`	Displays the system's configure date and time in the following format: day of the week; month; date time year; timezone. For example: *`Mon Mar 18 22:16:51 2002 GMT`*

Configuring Daylight Savings Rules

You use different commands to configure daylight savings rules depending on whether daylight savings at the appropriate location is:

- Nonrecurring (defined for a specific period of time). For example, the United States currently uses daylight savings rules that expire after 2006.
- Recurring (always occurs, with no defined stopping point). For example, the United States will start using recurring daylight savings rules in 2007.

> **Note**
> IPSO will automatically make this change for United States time zones in 2007.

Use the following commands to create daylight savings rules. You must enter a value for all the parameters to form a valid command.

```
add date timezone-dst location non-recurring
        start-year year
        start-month month
        start-date <1-31>
        start-time time
        end-year year
        end-month month
        end-date <1-31>
        end-time time
        dst-offset <00:00-24:00>

add date timezone-dst location recurring
        start-year year
        start-month month
        start-week occurrence
        start-day day
        start-time time
        end-month month
        end-week occurrence
        end-day day
        end-time time
        dst-offset <00:00-24:00>
```

Use the following commands to configure daylight savings rules. You do not have to enter a value for all the parameters to form a valid command.

```
set date timezone-dst location non-recurring rule start-year
        start-month month
        start-date <1-31>
        start-time time
        end-year year
        end-month month
        end-date <1-31>
        end-time time
        dst-offset <00:00-24:00>
```

```
set date timezone-dst location recurring rule start-year
        start-month month
        start-week occurrence
        start-day day
        start-time time
        end-month month
        end-week occurrence
        end-day day
        end-time time
        dst-offset <00:00-24:00>
```

Use the following commands to delete daylight savings rules.

```
delete date timezone-dst location
        non-recurring rule start-year
        recurring> rule

delete date timezone-dst location rules all
```

Use the following commands to view daylight savings rules.

```
show date timezone-dst location
        non-recurring rule start-year
        recurring> rule

show date timezone-dst location rules all
```

Arguments

location	Specifies a location in the time zone. For countries with multiple word names, such as the United States, you must bracket the location string with quotation marks, as in "United States/New_York." If you use single command mode, bracket multiple word location names with single quotation marks. See "Invoking the CLI" on page 22 for information about single command mode.
rule *start-year*	Specifies the rule you want to change by indicating its start year.
start-year *year*	Specifies the year in which the DST rule begins.
start-month *month*	Specifies the month when DST begins.
start-week *occurrence*	Specifies the occurrence of the relevant day in the month specified by start-month. For example, entering 2 for this parameter and entering Sun as the start-day specifies that DST will begin on the second Sunday of the specified month. The valid entries are 1, 2, 3, 4, and last.
start-date <1-31>	Specifies the day of month when DST begins.
start-day *day*	Specifies the day of week when DST begins. The valid entries are Sun, Mon, Tue, Wed, Thur, Fri, and Sat.
start-time *time*	Specifies the time when DST begins in 24-hour format.

`end-year` *year*	Specifies the year when the DST rule ends.
`end-month` *month*	Specifies the month when DST ends. The valid entries are `Jan`, `Feb`, `Mar`, `Apr`, `May`, `Jun`, `Jul`, `Aug`, `Sep`, `Oct`, `Nov`, and `Dec`.
`end-week` *occurrence*	Specifies the occurrence of the relevant day in the month specified by `end-month`. For example, entering `2` for this parameter and entering `Sat` as the `end-day` specifies that DST will end on the second Saturday of the specified month. The valid entries are `1`, `2`, `3`, `4`, and `last`..
`end-date <1-31>`	Specifies the day of month that DST ends.
`end-day` *day*	Specifies the day of week when DST ends. The valid entries are `Sun`, `Mon`, `Tue`, `Wed`, `Thur`, `Fri`, and `Sat`.
`end-time` *time*	Specifies the time when DST ends in 24-hour format.
`dst-offset <00:00-24:00>`	Specifies the amount by which the time is offset. You can use the following formats: • *hh* • *hh:mm*

Restoring the Default Rule

You cannot use the CLI to revert to the default daylight savings rule for a time zone. To configure the system to use the default rule, perform this procedure:

1. Log into the IPSO shell.

   In the CLI, you can enter `shell` to load the shell.

2. Delete the customized time zone file from /var/etc/zoninfo directory.

3. Mount / as read-write by entering

   ```
   mount -uw /
   ```

4. Change directory to /etc/zoneinfo/*region*. For example, for a United States time zone enter

   ```
   cd /etc/zoneinfo/America
   ```

5. Delete the symbolic link for the customized time zone.

6. Rename *time_zone*`.orig` as *time_zone*.

   For example, if you customized `New_York`, you would rename `New_York.orig` to `New_York`.

7. If you are restoring the default time zone rule for the currently selected time zone, perform these steps:

   a. Select a different time zone to be the current time zone.

   b. Reselect the original time zone.

Disk Commands

Use the commands in this section to show information about the hard drives in your appliance.

Viewing Disk Information

Use the following command to show the disks (by drive identification number) that IPSO detects on the local system:

```
show disks
```

Use the following commands to show information about a specified disk:

```
show disk
        id
        id model
        id type
        id capacity
        id geometry
        id location
```

Arguments

`disk` *id*	Specifies the drive identification number of the disks that IPSO detects. If you use this command without any additional arguments, the command displays the information in the rest of this table.
`model`	Specifies the model of the drive that IPSO detects.
`type`	Specifies whether the disk is a Bootmgr or IPSO disk.
`capacity`	Specifies the disk capacity in megabytes.
`geometry`	Specifies the logical block address geometry of the drive in CHS format: cylinders x heads x sectors per track detected for each drive.
`location`	Specifies the physical location of the drive on the chasis.

Disk Mirroring Commands

For platforms that support the feature, disk mirroring provides fault tolerance by letting your appliance continue working in the event of a disk failure. You can create mirror sets that consist of a source disk (which holds the active copy of the operating system) and mirror hard disk. The mirror disk contains a copy of all the files on the source disk, and if the source disk fails, the mirror disk immediately takes over. Your appliance continues to operate normally.

If you have an appliance on which you have configured disk mirroring, you can "warm swap" disk drives (with the exception of IP500 series appliances) —you can replace a drive without shutting down the appliance. This allows you to replace a failed drive without interrupting service.

Note
A mirror disk must be the same size or larger than the source disk. Before you create a mirror set, verify that this is true by using the `show disk` commands. See "Viewing Disk Information" on page 184 for information about these commands.

Configuring Disk Mirroring

Use the following command to add a disk mirror set:

```
add diskmirror
```

Use the following command to delete a disk mirror set. You cannot delete a disk mirror set until the synchronization is 100 percent.

```
delete diskmirror id
```

Use the following command to view the identification number of a disk mirror set on your system:

```
show diskmirrors
```

Use the following commands to view properties about disk mirroring on your system:

```
show diskmirror
        id
        id mrdrive
        id srcdrive
        id syncpercent
```

Arguments

id	Specifies the disk mirror set identification number. If you use this command without any additional arguments, the command displays all of the rest of the information in this table.
mrdrive	Displays the disk ID of the mirror drive.
srcdrive	Displays the disk ID of the source drive
syncpercent	Displays the percentage of sync zones that are currently synchronized. Enter this command repeatedly to see updated percentage figures.

NTP

Use the commands in this section to configure Network Time Protocol (NTP) settings for your system.

Configuring NTP

Use the following commands to specify other systems as network time protocol servers or peers for this system:

```
add ntp
        server ip_address version <1-3> [prefer <yes | no>]
        peer ip_address version <1-3> [prefer <yes | no>]
```

Arguments

server *ip_address*	Specifies the address of a time server from which this machine synchronizes its clock. The specified server does not synchronize its clock to the local clock of this system.

`peer ip_address`	Specifies the address of a time server with which this machine synchronizes clocks. The specified server can synchronize its clock to the local clock of this system.
`version <1-3>`	Specifies which version of NTP to use when synchronizing with the specified system. Nokia recommends that this be set to version 3, the most recent version.
`prefer <yes | no>`	Specifies whether to select this system as the time source if more than one server or peer is available to another system that requests a time source.This setting is used as a tiebreaker. **Default:** no

Use the following commands to configure network time protocol settings and to configure this system as a master NTP server:

```
set ntp
        active <on | off>
        server ip_address version <1-3> [prefer <yes | no>]
        peer ip_address version <1-3> [prefer <yes | no>]
        master <yes | no>
        stratum <1-15> source local-clock
```

Arguments

`active <on | off>`	Specifies whether the time service is active or inactive. When NTP is active, the local clock will be synchronized as configured and other systems will be able to set their time from this system.
`server ip_address`	Specifies the NTP server that you want to configure.
`peer ip_address version`	Specifies the NTP peer that you want to configure.

`version <1-3>`	Specifies which version of NTP to use when synchronizing with the specified system. It is recommended that this be set to version 3—the most recent version.
`prefer <yes | no>`	Specifies whether this system should be selected as the time source if more than one server or peer is available to another system that requests a time source.This setting is used as a tie-breaker. **Default:** no
`master <yes | no>`	Configures this system to act as an NTP master server. When configured as a master server, a system will not get its time from other systems.
`stratum <0-15>` `source local-clock`	Specifies the stratum—the number of hops away from a source of correct time this system's clock should is. This should normally be set to 0. **Default:** 0

Use the following commands to stop a system from using NTP to synchronize with other systems that it was previously configured to synchronize with:

```
delete ntp
        server ip_address
        peer ip_address
```

Arguments

`server` *`ip_address`*	Specifies the NTP server to prevent this system from synchronizing with.
`peer` *`ip_address`*	Specifies the NTP peer to prevent this system from synchronizing with.

Use the following commands to view the NTP configuration settings for this system:

```
show ntp
        active
        servers
        peers
        <server | peer> ip_address version [prefer]
        master
```

Arguments

`active`	Shows whether or not NTP is active.
`servers`	Lists any systems configured as NTP servers and shows the appropriate NTP version and `prefer` settings.
`peers`	Lists any systems configured as NTP peers and shows the appropriate NTP version and `prefer` settings.
`<server | peer>` *`ip_address`* `version [prefer]`	Shows the NTP version (and `prefer` setting if specified) of the specified server or peer.
`master`	Shows whether this system has been configured as an NTP master server. If it has, this command also shows the appropriate `stratum` setting.

Package Commands

Use the commands in this section to install, upgrade, and delete packages and to view information about packages on your appliance.

Managing Packages

Use the following command to show information about packages installed on the local system:

```
show package
        all
        active
        inactive
```

Arguments

`all`	Lists both the active and inactive packages installed on the system.
`active`	Lists the active packages installed on the system.
`inactive`	Lists the inactive packages installed on the system.

Use the following commands to show a specific package or all packages in a specified directory on a remote or local system. The packages are stored in a gnu zipped tar file with a *.tgz file extension.

```
show package media
        ftp addr ip_address user name password password dir name
        anonftp addr ip_address dir name
        cdrom dir name
        local dir name
```

Arguments

`addr` *`ip_address`*	Specifies the IPv4 address of the remote machine containing the package. Example: 192.168.10.10
`user` *`name`*	Specifies the login name for FTP.

`password` *`password`*	Specifies the password associated with the username parameter for FTP login.
`dir` *`name`*	Specifies the full path of the directory on the remote or local system that contans the packages. Example: `/opt/packages`

You can add optional packages to the core system software. The contents of the package must conform to the predefined IPSO directory hierarchy in order for the package to become integrated. The valid suffixes are tzg, tar.gz, tar, and tar.Z. Each package will be installed as a subdirectory of `/opt`.

Use the following commands to add a package located on a remote system or local system:

```
add package media
        ftp addr ip_address user name password password name name
        anonftp addr ip_address name name
        cdrom name name
        local name name
```

Arguments

`addr` *`ip_address`*	Specifies the IPv4 address of the remote machine containing the package. Example: 192.168.10.10
`user` *`name`*	Specifies the login name for FTP.
`password` *`password`*	Specifies the password associated with the username parameter for the FTP login.
`name` *`name`*	Specifies the file name of the package to install. Use the complete path. Example: `/opt/packages/IPSO-3.7.tgz`

Use the following commands to upgrade the existing package (*.tgz) by specifying a different package located on a remote or local system:

```
upgrade package media
        ftp addr ip_address user name password password old name new name
        anonftp addr ip_address old name new name
        cdrom old name new name
        local old name new name
```

Arguments

addr *ip_address*	Specifies the IPv4 address of the remote machine containing the package. Example: 192.168.10.10
user *name*	Specifies the login name for FTP.
password *password*	Specifies the password associated with the username parameter for FTP login.
old *name*	Specifies the name of the existing package to be replaced. Use the complete path.
new *name*	Specifies the name of the package (in .tgz format) you will use to replace the existing package. Use the complete path.

Use the following command to activate or deactivate a specified package:

```
set package name name <on | off>
```

Use the following command to uninstall a specified package:

```
delete package name name
```

Arguments

name *name*	Specifies the name of the package. Use the complete path.

Advanced System Tuning Commands

The commands in this section are intended for very specific purposes, and, under most circumstances, you should not change any of the default settings.

Controlling Sequence Validation

Use the following command to enable and disable sequence validation:

```
set advanced-tuning tcp-options sequence-validation <on | off>
```

Use the following command to view whether sequence validation is enabled or disabled:

```
show advanced-tuning tcp-options sequence-validation
```

Tuning the TCP/IP Stack

Use the following command to set the TCP maximum segment size (MSS) for segments received by your local system:

```
set advanced-tuning tcp-ip tcp-mss <512-1500>
```

The default value is 1024.

Use the following command to view the configured TCP MSS value:

```
show advanced-tuning tcp-ip tcp-mss
```

Router Alert IP Option

Use the following command to specify whether IPSO should strip the router alert IP option before passing packets to the firewall. (The router alert IP option is commonly enabled in IGMP packets.)

```
set advanced-tuning ip-options stripra <1 | 0>
```

Use the following command to view the configured setting:

```
show advanced-tuning ip-options stripra
```

IP1260 Port Optimization

You can use the following command to optimize the performance of the interfaces of two-port Gigabit Ethernet NICs in IP1260 platforms when the interfaces forward unidirectional UDP traffic.

```
set advanced-tuning ethernet-options <on | off>
```

Enabling this option does not optimize throughput for other types of traffic or other interfaces. This command is not available on the IP1220.

Caution
Do not enable this option if more than two Gigabit Ethernet interfaces are installed in the system. Doing so can impair system performance.

4 High Availability Commands

Nokia provides the following solutions that you can use to create a highly available and redundant configuration to ensure that your network traffic continues to flow in the event that one of your firewall platforms fails:

- IP clustering
- External load balancer support
- Single license VRRP (in which one firewall license is shared between two systems). There is no CLI command for this feature.
- Multiple license VRRP (in which each system has an individual firewall license installed)

For information about the CLI commands to use with an IP cluster, see the *Chapter 5, "IP Clustering Commands."* The commands for external load balancer support and multiple license VRRP are described in this chapter.

External Load Balancer Command

You can use an external load balancer to balance traffic to multiple IPSO firewalls without using IP clustering or VRRP. By configuring the firewalls to synchronize traffic with each other you can provide high availability as well. Using an external load balancer also has the advantage of not requiring you to use virtual IP addresses on the IPSO firewalls.

Use the following command to enable or disable support for and external load balancer:

```
set external_load_balancer <on | off>
```

VRRP Commands

You can configure the Virtual Router Redundancy Protocol (VRRP) to use either monitored-circuit VRRP or VRRPv2. You can configure monitored-circuit VRRP using either the simplified method or the full method. For more information, refer to the *Network Voyager Reference Guide*.

Note
Beginning with IPSO 3.8.1, Nokia also supports VRRP for IPv6 addresses. For more information about the CLI commands for this implementation, see "VRRP for IPv6" page 272.

The CLI commands for these implementations are explained in the following sections.

General VRRP Commands

Use this group of commands to set and view parameters that apply to any VRRP configuration, regardless of which VRRP implementation you use.

```
set vrrp
        accept-connections <on | off>
        coldstart-delay seconds
        monitor-firewall <on | off>
        monitor-hdd <on | off>
```

Arguments

`accept-connections <on | off>`	The VRRP protocol specifies that a router should not accept or respond to IP packets sent to an adopted VRRP (virtual) backup IP address. Entering `off` specifies compliance with the specification. Entering `on` overrides this behavior and allows the master to accept and respond to packets sent to an adopted VRRP backup address. This setting enhances interaction with network management tools and allows you to log into the VRRP master using a backup address. You must enable this option when deploying dynamic routing protocols or any highly available application whose service is tied to a VRRP backup address. **Default:** off
`coldstart-delay` *`seconds`*	Specifies a number of seconds that the system should wait after starting before joining a VRRP group. You might want to configure a delay to allow routing adjacencies to form or for applications to synchronize before a system becomes the VRRP master. You can use this option with or without firewall monitoring. If you also enable firewall monitoring, the system begins to monitor the firewall after the coldstart delay period has elapsed. **Default:** 0 seconds
`monitor-firewall <on | off>`	Specifies whether to monitor the state of the firewall and respond appropriately. If a VRRP master detects that the firewall is not ready to handle traffic or is not functioning properly, the master fails over to a backup system. If all the firewalls on all the systems in the VRRP group are not ready to forward traffic, no traffic will be forwarded. **Default:** on

`monitor-hdd <on | off>`	Specifies whether virtual routers should transition to init state (and a failover should occur) if certain disk errors are detected on the master. If you enable this option on a system on which disk mirroring is also enabled, virtual routers do not transition to init state if the mirror disk takes over for the primary disk. **Default:** off

Simplified Method Monitored-Circuit VRRP

Use the commands explained in this section to configure monitored-circuit VRRP implementations using a simplified method. When you use this method, you create backup (virtual) addresses and the system automatically associates the appropriate router interfaces with the backup addresses. This reduces the number of configuration steps you need to perform.

> **Note**
> You cannot convert legacy monitored-circuit configurations into a simplified configuration. To use this method, you must first delete any existing legacy monitored-circuit configuration.

Use the following commands to create a virtual router:

```
add mcvr vrid <1-255> priority <1-254> priority-delta <1-254>
        authtype <none|simple> [password passwd]
        hello-interval <1-255>
```

Use the following commands to add backup addresses to a virtual router:

```
add mcvr vrid <1-255> backup-address ip_address
        vmac-mode <default-vmac|extended-vmac|interface-vmac|static-
           vmac static-mac static_VMAC>
```

Use the following commands to configure, view, and delete virtual routers:

```
set mcvr vrid <1-255>
        authtype <none|simple> [password passwd]
        hello-interval <1-255>
        priority <1-254>
        priority-delta <1-254>

show mcvr vrid <1-255>
        all
        authtype
        backup-addresses
        hello-interval
        password
        priority
        priority-delta

show mcvr vrids

delete mcvr
        old-mc-config
        vrid <1-255>
               backup-address ip_address
```

You can also enable and disable preempt mode after you configure monitored-circuit VRRP using the simplified method. See "Full Method Monitored-Circuit VRRP" for information about how to do this. The command described in that section also works for simplified monitored-circuit configurations.

Arguments

`mcvr vrid <1-255>`	Specifies a virtual router ID. The ID must be unique on the network that its backup addresses belong to. The ID must be identical on each physical router that participates in the virtual router.
`priority <1-254>`	Specifies or shows this (physical) router's priority during contention for a failed router's addresses. **Default:** 100

`priority-delta <1-254>`	If an interface associated with a backup address fails, the value of the priority delta is subtracted from the priority to yields an effective priority for the physical router. When the effective priority on the master is less than the priority of another router in the VRRP group, a new master is selected.
`authtype <none | simple> [password` *`passwd`*`]`	Specifies or shows whether to use authentication. To use authentication, select simple and enter a password 1 to 8 characters in length. The authentication type and password (if any) must be identical on each physical router participating in the virtual router. **Default:** none
`hello-interval <1-255>`	Specifies or shows the interval in seconds between VRRP advertisements. This value must be the same on all routers participating in the virtual router. **Default:** 1 second
`backup-address` *`ip_address`*	Specifies or shows an IP address for the virtual router. This address must be on the same network as one of the interfaces in the physical router but must not match a real IP address of any device on this network. You must configure the same backup address on each physical router participating in the virtual router.
`vmac-mode <default-vmac | extended-vmac | interface-vmac | static-vmac` *`static_VMAC`*`>`	Specifies or shows how the virtual MAC address for the backup address is created. For more information on the options, refer to the *Network Voyager Reference Guide*. **Default:** default-vmac
`mcvr vrids`	Shows all attributes of all the virtual routers configured on the system.
`all`	Shows all the attributes of the specified virtual router.

`old-mc-config`	Deletes any monitored-circuit configuration that was created using the full method of VRRP configuration. You must delete these configuration before you create a monitored-circuit configuration using the simplified method.

Full Method Monitored-Circuit VRRP

Use these commands configure properties for specific interfaces for the monitored circuit implementation of VRRP.

```
set vrrp interface if_name monitored-circuit vrid <1-255>
        monitored-interface if_name <on | off>
        monitored-interface if_name priority delta <1-254>
        auto-deactivation <on | off>
        priority <1-254>
        hello-interval <1-255|default>
        vmac-mode <default-vmac|extended-vmac|interface-vmac|static-
           vmac mac_address>
        backup-address ip_address <on | off>
        preempt-mode <on | off>
```

The following section explains the use and meaning of VRRP monitored circuit commands.

Arguments

`monitored-interface` *`if_name`* `<on | off>`	Specifies the ID for a virtual router with monitored circuit dependencies and the associated interface.
`monitored-interface` *`if_name`* `priority-delta <1-254>`	Specifies the priority delta associated with the interface with a dependency on the virtual router.

`monitored interface if_name auto-deactivation <on | off>`	Specifies to allow the effective priority to go to 0 and for the virtual router to be removed from the network. In the typical implementation, if the effective priority goes to 0, the protocol reestablishes a value of 1. **Default:** off
`priority <1-254>`	Specifies the priority assigned to the virtual router during contention for a fail router's addresses.
`hello-interval <1-255>`	Specifies the interval in seconds between VRRP advertisements. This value should be the same on all the routers participating in the virtual router. **Default:** 1 second
`vmac-mode <default-vmac | extended-vmac | interface-vmac | static-vmac mac_address>`	Specifies the method to use to set the virtual MAC address for the specified virtual router. For information on the options, see the *Network Voyager Reference Guide*.
`backup-address <ip_address> <on | off>`	Specifies for the user to enter a backup IP address and enable or disable it.
`preempt-mode <on | off>`	Set to On to specify that this router will not fail over to a router with higher priority. Use this setting if you want to reduce the number of transitions.

VRRP Show Commands

Use the following commands to monitor and troubleshoot your VRRP implementation.

```
show vrrp
        interfaces
        interface if_name
        stats
        summary
```

VRRPv2

Use the commands explained in this section to configure VRRPv2 implementations.

```
set vrrp interface if_name
        off
        authtype <none|simple password>

set vrrp interface if_name virtual-router vrid <1-255>
        <on | off>
        hello-interval <1-255|default>
        vmac-mode <default-vmac|extended-vmac|interface-vmac|static-
           vmac mac_address>
        backup-address ip_address <on | off>

set vrrp interface if_name virtual-router backup-vrid <1-255>
        <on | off>
        backup-address ip_address <on | off>
        hello-interval <1-255|default>
        preempt-mode <on | off>
        priority <1-254>
        vmac-mode <default-vmac|extended-vmac|interface-vmac|static-
           vmac mac_address>
```

The following section explains the use and meaning of VRRP version 2 commands.

Arguments

`off`	Specifies to disable VRRP on the specifed interface.

`<on | off>`	Specifies the virtual router ID for the virtual router used to backup the local interface's address(es). The VRID must be unique for all virtual routers running on the interface's network. Enter off to remove the specified virtual router.
`hello-interval <1-255>`	Specifies the interval in seconds between VRRP advertisements. This value must be the same on all the routers participating in the virtual router. **Default:** 1 second
`authtype <none|simple password>`	None specifies not to use any authentication. Simple specifies to use simple password authentication. Enter plain text between 1 and 8 characters long. This password applies to all the virtual routers configured on an interface.
`vmac-mode <default-vmac | extended-vmac | interface-vmac | static-vmac mac_address>`	Specifies the method to use to set the virtual MAC address for the specified virtual router. For information on the options, see the *Network Voyager Reference Guide*.
`backup-address` *`ip_address`* `<on | off>`	Specifies a virtual router ID for the virtual router used to backup another system's IP address(es). The router you are backing up must also have this virtual router configured for its addresses. Enter an IP address to assign to the virtual router used to backup another system's IP addresses.
`priority <1-254>`	Specifies this physical router's priority during contention for a failed router's addresses.
`hello-interval <1-255 | default>`	Specifies the interval in seconds between VRRP advertisements. This value must be the same on all routers participating in this virtual router. **Default:** 1 second

`preempt-mode <on | off>`	Set to on to specify that this router will not fail over to a router with higher priority. Use this setting if you want to reduce the number of transitions. This parameter is only available if the virtual IP address is the same as the interface IP address.

5 IP Clustering Commands

This chapter describes the commands you use to configure clustering on your system and to view current settings.

A cluster is a group of IPSO systems that appear as a single system to devices outside the cluster. IP traffic sent to the cluster IP address is load balanced between the cluster members, and the cluster continues to function if a member fails or is taken out of service for maintenance purposes.

Use the commands in this section to add, configure, and delete clusters.

See the *Voyager Reference Guide* for information about how to configure and manage an IPSO cluster.

Note
The IP2250 and IP2255 platforms do not support IP clustering.

General Clustering Commands

Use the following commands to create a cluster configuration.

`add cluster id <0—65535> [passwd passwd]`

```
add cluster id <0–65535>
        feature name
        interface log_if_name cluster-address ip_address
        network network/mask cluster-address ip_address

add cluster
        ip-pool network network/mask member ip_address
        vpn-tunnel network ip_address/mask destination ip_address
```

Arguments

`id <0–65535> [passwd` *`passwd`*`]`	Creates a cluster configuration and specifies its unique identification number. If you enter this command without specifying a password, the system responds with `Enter password for cadmin :` If this is the first member of the cluster, create a password for the cadmin user by entering it now. The password must have at least six characters. If this is not the first member of the cluster, enter the cadmin password that was used on the other members. When you enter the password, you are asked to enter it again to verify it.
`feature` *`name`*	Specifies a feature that should be shared when a system joins the cluster. These are called *join-time shared features.* This command is valid only if you have removed a feature from the list of join-time shared features and want to make it shared again. See "Managing Join-Time Shared Features" for more information.
`interface` *`log_if_name`*	Adds an Ethernet interface to a cluster.

`network` *network/mask*	Specifies a network to be added to the cluster. You must specify the appropriate subnet mask. The interface that is configured with an address in the specified network is added to the cluster.
`cluster-address` *ip_address*	Specifies the cluster IP address for this interface. The cluster IP address is shared by all the cluster interfaces on a given network. When specified for an interface, the cluster address must belong to one of the networks with which the interface is configured.
`ip-pool network` *network/mask* `member` *ip_address*	Specifies a range of addresses to use as an IP pool. You must specify which cluster member should manage the specified addresses by entering the real IP address of the primary cluster protocol interface of the appropriate member. Use this command to specify IP pool addresses used with tunnels formed with non-Check Point gateways or clients. If the other end of the tunnel is a Check Point gateway, do not use this command—simply specify the IP pool using VPN-1 NG AI.
`vpn-tunnel network` *network/mask* `destination` *ip_address*	Creates one end of a VPN tunnel. Use `network` *network/mask* to specify the IPv4 network address and mask of the remote encryption domain. Use `destination` *ip_address* to specify the IPv4 address of the remote tunnel endpoint. Use this command if the other end of the tunnel is a non-Check Point gateway or client. If the other end of the tunnel is a Check Point gateway, do not use this command—simply specify the tunnel using VPN-1 NG AI.

Use the following commands to configure properties for an existing cluster:

```
set cluster id <0–65535>
        cadmin passwd oldpass passwd newpass passwd
        change <0–65535>
        coldstart-delay integer
        failure-interval integer
        firewall-check-required <yes | no>
        interface log_if_name
              cluster-address ip_address
              hash <default | on-destination-ip | on-source-ip>
        join-remote ip_address
        mode <mcast | mcast-group | forwarding | unicast>
        network network/mask cluster-address ip_address
        performance-rating <0–65535>
        primary-interface log_if_name
        primary-network network/mask
        remote-node ip_address performance-rating integer
        secondary-interface log_if_name
        secondary-network network/mask
        state <up | down>
        work-assign <static | dynamic>

set cluster
        ip-pool network network/mask member ip_address
        securemote <yes | no>
        vpn-clients <yes | no>
        vpn-interop <yes | no>
        vpn-tunnel network network/mask destination ip_address
```

Arguments

`id <0–65535>`	Specifies the unique identification number of an existing cluster.

`cadmin passwd oldpass` *`passwd`* `newpass` *`passwd`*	Specifies a new password for the cadmin user (the cluster adminstrator user). You must include the current (old) password as well as the new password. The new password must have at least six characters. See "Clustering Administration" for information about the cadmin user.
`change <0–65535>`	Specifies the new cluster identification number.
`coldstart-delay <20–200>`	Specifies the number of seconds the system waits before starting the cluster protocol. This allows VPN-1/FireWall-1 to become active and synchronize before the cluster protocol starts. VPN-1/FireWall-1 NG_AI does not require this delay. **Default:** 30
`failure-interval <500–10000>`	Specifies the number of milliseconds the system waits before assuming the cluster has dissolved . If the specified time passes without the member receiving cluster protocol keep-alive messages, the member leaves the cluster and attempts to rejoin. The range is 500 through 10000. **Default:** 500

`firewall-check-required <yes | no>`	Specifies whether this system should become a member of a cluster only if VPN-1/FireWall-1 is running. This option also specifies whether IPSO should monitor VPN-1/FireWall-1 and remove the member from the cluster if the firewall stops functioning. If VPN-1/FireWall-1 is not running at the time you change the cluster state to up, set this option to no temporarily. In this case, be sure to set the option to yes before you put the cluster into service (assuming that you are using FireWall-1). If VPN-1/FireWall-1 is not running and you do not disable firewall monitoring, the member cannot be part of a cluster (even if it is the only member). Be sure to enable firewall monitoring before you put the cluster into service.
`interface` *`log_if_name`*	Specifies a logical interface name.
`cluster-address` *`ip_address`*	Specifies a cluster IP address for this interface or network. The cluster IP address is shared by all the cluster interfaces on a given network. When specified for an interface, the cluster address must belong to one of the networks with which the interface is configured.

`hash <default | on-destination-ip | on-source-ip>`	Use the hash option to configure the manner in which IPSO balances traffic among cluster nodes. If you do not use NAT in the protected networks, use the default option. One node will handle incoming and outgoing traffic for a given connection. Use the default option if you use NAT in the protected networks and want the cluster to support asymmetric connections. If you select this option, IPSO uses the source and destination IP addresses as inputs to its load balancing mechanism. Because the addresses are changed by NAT, the cluster might split the connection between two nodes. If you are choosing the hash method for an interface that uses NAT, and the destination interface also uses NAT, use the default hash method for the interfaces at both ends of the link. The other hash options use only one IP address (source or destination) as inputs to the load balancing mechanism. Use these options if you use NAT in the protected networks and want to force connections to be symmetric. • For external interfaces, use the on-source-ip. • For internal interfaces, use on-destination-ip.

`join-remote` *ip_address*	Specifies that the system you are logged into should become a member of an existing cluster by *joining*. When joining a cluster, a system copies a variety of configuration settings from another cluster member (so you don't have to configure these settings manually). Specify an IP address of an existing cluster member that this system should copy configuration settings from. Follow these guidelines when specifying the IP address: • The address should be assigned to an interface that belongs to the cluster master. • The interface must be one of the master's cluster interfaces. • You should use the "real" address of the interface—not its cluster IP address.
`mode <mcast | mcast-group | forwarding | unicast>`	Specifies the clustering mode. All cluster members must use the same mode. Use `mcast-group` if the cluster is connected to switches that are using IGMP snooping. This configuration restricts the clustering protocol traffic to only the cluster nodes.
`network` *network/mask*	Specifies a network connected to one of the cluster interfaces. You must specify the appropriate subnet mask.
`performance-rating <0–65535>`	Specifies the performance rating for this member.

`primary-interface` *`log_if_name`*	Specifies the primary cluster protocol interface. Cluster members use this interface to exchange cluster protocol messages with the other cluster members. For security reasons this interface should be an internal interface.
`primary-network` *`network/mask`*	Specifies the primary cluster protocol network. Cluster members exchange cluster protocol messages over this network. Each member must use the same primary cluster protocol network.
`remote-node` *`ip_address`* `performance-rating` *`integer`*	Specifies the performance rating for another cluster member. You must specify the IP address of a cluster interface on the other member. You can perform this command only if you have logged in as cadmin (a cluster administor). See "Clustering Administration" for information about the cadmin user.
`secondary-interface` *`log_if_name`*	Specifies the (optional) secondary cluster protocol interface. Cluster members use this interface to exchange cluster protocol messages with the other cluster members if their primary cluster interface fails. For security reasons this interface should be an internal interface.
`secondary-network` *`network/mask`*	Specifies the (optional) secondary cluster protocol network. Cluster members exchange cluster protocol messages over this network if the primary cluster protocol network fails. Each member must use the same secondary cluster protocol network.

`state <up | down>`	Configures the cluster state. The cluster state can be set to `up` only if: • A primary interface is selected. • The cluster has two interfaces configured with cluster IP addresses. • All dynamic routing protocols and routing services are disabled. • The member is configured with a valid performance rating.
`work-assign <static | dynamic>`	Specifies whether the cluster can rebalance the load of active connections by moving them between nodes. • `static` prevents the cluster from moving active connections between nodes.Use for Check Point applications and features that require “bidirectional stickiness,” which means that all the packets for a given connection must be processed by the same node. Also use if you are using IP pools with non-Check Point gateways or clients. • `dynamic` allows the cluster to periodically rebalance the load by moving active connections between nodes. Use for optimum load balancing.

`ip-pool network` *`network/mask`* `member` *`ip_address`*	Specifies a cluster member to manage the IP pool specified by `network` *`network/mask`*. range of addresses to use as an IP pool. *`ip_address`* must be the real IP address of the primary cluster protocol interface of the member that should manage the pool of addresses. Use this command to specify IP pool addresses used with tunnels formed with non-Check Point gateways or clients. If the other end of the tunnel is a Check Point gateway, do not use this command—simply specify the IP pool using VPN-1 NG AI.
`securemote <yes | no>`	Specifies whether SecuRemote clients can connect to the system. **Default:** no
`vpn-clients <yes | no>`	Specifies whether the cluster supports VPNs with non-Check Point clients.
`vpn-interop <yes | no>`	Specifies whether the cluster supports VPNs with non-Check Point gateways.
`vpn-tunnel network` *`network/mask`* `destination` *`ip_address`*	Specifies the end of a VPN tunnel formed with a non-Check Point gateway. Use `network` *`network/mask`* to specify the IPv4 network address and mask of the remote encryption domain. Use `destination` *`ip_address`* to specify the IPv4 address of the remote tunnel endpoint (non-Check Point gateway).

Use the following commands to delete a cluster or to turn off specified features:

```
delete cluster id <0—65535>
```

```
delete cluster id <0–65535>
        feature feature
        interface log_if_name
        network network/mask
        secondary-interface log_if_name
        secondary-network network/mask

delete cluster
        ip-pool network network/mask
        vpn-tunnel network network/mask
```

Arguments

`id <0–65535>`	Specifies the unique identification number of the cluster.
`feature` *feature*	Specifies a feature to remove from the list of join-time shared features.
`interface` *log_if_name*	Specifies the logical name of an interface to disassociate from the cluster. You cannot delete the primary cluster protocol interface.
`network` *network/mask*	Specifies a network to disassociate from the cluster. You cannot delete the primary cluster protocol network.
`secondary-interface` *log_if_name*	Reconfigures the cluster so that the specified interface is no longer the secondary cluster protocol network. This command does not disassociate the interface from the cluster.
`secondary-network` *network/mask*	Reconfigures the cluster so that the specified network is no longer the secondary cluster protocol network. This command does not disassociate the network from the cluster.
`ip-pool network` *network/mask*	Deletes the specified IP pool.

`vpn-tunnel network` *`network/mask`*	Deletes the specified VPN tunnel.

Use the following commands to view various information about IPSO clusters:

```
show clusters

show cluster id <0—65535>
        coldstart-delay
        failure-interval
        features
        firewall-check-required
        info
        interfaces
        interface log_if_name cluster-address
        member info
        mode
        network network/mask cluster-address
        networks
        performance-rating
        primary-interface
        proto-state
        remote-node ip_address performance-rating
        secondary-interface
        secondary-network
        state
        work-assign

show cluster
        ip-pools
        securemote
        securemote clients
        vpn-clients
        vpn-interop
        vpn-tunnels
```

Arguments

`clusters`	Shows summary information a cluster configured on this system. If you logged in as admin, this command also shows a variety of cluster-related information about the member you logged into. If you logged in as cadmin, this command also shows a variety of information about each member of the cluster. See "Clustering Administration" for information about the cadmin user.
`coldstart-delay`	Shows the number of seconds the system waits before starting the cluster protocol.
`failure-delay`	Shows the number of milliseconds the system waits before assuming the cluster has dissolved.
`info`	Shows all the configuration and monitoring information for the specified cluster.
`features`	Shows the join-time shared features.
`firewall-check-required`	Shows whether the system will wait for VPN-1/ FireWall-1 to start before it becomes a member of a cluster.This command also shows whether IPSO will monitor VPN-1/FireWall-1 and remove the member from the cluster if the firewall stops functioning.
`interfaces`	Shows the logical names of all the cluster interfaces.
`interface` *`log_if_name`* `cluster-address`	Shows the cluster IP address for the specified interface.

`member info`	If you logged in as admin, this command shows a variety of cluster-related information about the member you logged into. If you logged in as cadmin, this command shows a variety of information about each member of the cluster.
`mode`	Shows the clustering mode.
`network` *`network/mask`* `cluster-address`	Shows the cluster IP address for the specified network.
`networks`	Shows all the networks in which this cluster is participating.
`performance-rating`	Shows the performance rating of this member.
`primary-interface`	Shows logical name of the primary cluster protocol interface for the cluster.
`proto-state`	Shows the cluster protocol state (master, member, or uninitialized).
`remote-node` *`ip_address`* `performance-rating`	Shows the performance rating of the member specified by *`ip_address`*, which must be an IP address of the primary cluster protocol interface of one of the cluster members (including the member you are logged into).
`secondary-interface`	Shows the logical name of the secondary cluster protocol interface for the cluster.
`secondary-network`	Shows the secondary cluster protocol network for the cluster.
`state`	Shows the cluster state (up or down).
`work-assign`	Shows the work assignment method.

`ip-pools`	Shows the configuration of any IP pools used with VPN tunnels formed with non-Check Point gateways or clients.
`securemote`	Shows whether SecuRemote client access is enabled.
`securemote clients`	Shows whether there are any SecuRemote clients connected.
`vpn-clients`	Shows whether the cluster supports VPNs with non-Check Point clients.
`vpn-interop`	Shows whether the cluster supports VPNs with non-Check Point gateways.
`vpn-tunnels`	Shows the configuration of any VPN tunnels formed with non-Check Point gateways or clients.

Clustering Administration

Note
See the *Voyager Reference Guide* for information about how to configure and manage an IPSO cluster.

If you log into command-line session with cadmin priviledges (for example, if you use the user name cadmin), you are logged in as a cluster administrator. The prompt indicates this by showing `CCLI` and indicating the cluster ID. For example, the following prompt is for a cluster with the cluster ID 10:

NokiaCCLI:173 Cluster(10)>

(If there is no cluster configuration on a system, a cadmin user has not been created and you cannot log into the system as a cadmin user.)

As a cluster administrator, you can change and view configuration settings on all the cluster members in one command-line session.

Note
While logged in as cadmin, you can use all the clustering CLI commands. Instead of being applied to one member, the commands are applied to all the members.

A cluster administrator can configure each of the cluster members to use the same configuration settings for most clustering-related features. For example, if you are logged in as cadmin and enter

```
set cluster id 10 coldstart-delay 40
```

the coldstart delay is set to 40 seconds on all the cluster members.

Some cluster settings are not appropriate for being configured identically on all the members. For example, you cannot change the IP address of interfaces using the CCLI because interfaces on different members have different IP addresses. You can change cluster IP addresses because these must be consistent on all the members.

Note
If you are logged in as cadmin and enter a command that is not available to a cluster administrator, the CLI responds that the command is invalid.

Managing Join-Time Shared Features

Note
See the *Voyager Reference Guide* for complete information about join-time shared features.

You may want to have many configuration settings be identical on each cluster node. Voyager makes this easy for you by letting you specify which features will be configured the same on all cluster nodes. The features that are configured this way are called *join-time shared features*. Their configurations are shared when:

- a node joins (or rejoins) the cluster
- a new master is selected because the original master has left the cluster (for example, if it was rebooted)

In addition to helping you make sure that all cluster members are configured consistently, using this feature makes the configuration process easier and faster.

To see the list of features that are shared at join time, enter

```
show cluster id integer features
```

To remove a feature from this list so that its configuration information is not copied to a system when the system joins a cluster, enter

```
delete cluster id integer feature feature
```

Note
To ensure that cluster members are configured identically, you should avoid deleting features from the list of join-time shared features after the cluster is operational.

Configuring Join-Time Shared Features

When you log in as cadmin and change a setting of a join-time shared feature, the change is made on all the members. If a system later joins the cluster, it copies the modified settings for this feature.

To configure the settings of join-time shared features, you use the same CLI commands as an admin user.To learn these commands, see the appropriate sections in this guide. For example, to configure ARP entries for a cluster, see the chapter on configuring interfaces.

Changes made to the configuration settings of shared features overwrite any conflicting settings made by someone logged into an individual member as admin. For example, assume that DNS is a shared feature and an admin user sets the domain name on one member to foo.bar.com. If you log in as cadmin and change the domain name to your.company.com, the new name replaces foo.bar.com.

However, nonconflicting changes made as admin on an individual member are not overwritten. For example, if an admin user configures a static route on a member and you later configure a static route as cadmin, the new route is added to the original route.

If you remove a feature from the list of join-time shared features, you can still configure this feature while logged in as cadmin. The change is made on all the members, but systems that join the cluster later do not copy the configuration settings for that feature. You see a message that alerts you to the fact that systems join later will not copy this setting.

If you log into a member as admin and change a setting of join-time shared feature, the change is implemented on the system you are logged into but not implemented on the other members. Nokia recommends that you do not make changes to cluster settings or cluster shareable features on individual members—log in as cadmin to make these changes.

Some settings of join-time shared features cannot be configured using the CCLI. For example, you cannot set SSH host and identity keys. To configure these settings, you must log into the individual cluster members as admin.

Installing IPSO Images on a Cluster

As cadmin, you can upgrade the IPSO image on all the cluster members using one CLI session. (See "Managing IPSO Images" for information about upgrading images.)

After the new image has been successfully installed on all the members, you need to reboot them so that they will run the new image. Use the following commands to reboot cluster members:

```
reboot

reboot
        image <name | last-download>
              cluster-all
              cluster-force
              save
```

Arguments

`reboot`	Reboots each of the cluster members members in a staggered manner so that only one member is out of service at a time. Used by itself, `reboot` will reboot the cluster members with the image they are running prior to the reboot.
`image <name | last-download>`	Reboots the cluster members with the specified IPSO image. The last-download argument specifies to use the image most recently downloaded. The members are rebooted in a staggered manner.

`cluster-all`	Use this command to reboot all the cluster members simultaneously. You will be prompted to verify that you want to reboot all the cluster members. If you reboot a cluster this way, there will be an interruption in service while all the members are rebooting.
`cluster-force`	Use this command to reboot all the cluster members simultaneously. You will not be prompted to verify that you want to reboot all the cluster members. If you reboot a cluster this way, there will be an interruption in service while all the members are rebooting.
`save`	Saves any unsaved configuration changes prior to booting.

6 SNMP Commands

This chapter describes the SNMP configuration commands that you can enter from the initial CLI prompt, called Command mode.

SNMP Description

Use this group of commands to set and view parameters for SNMP. Through the SNMP protocol, network management applications can query a management agent using a supported MIB. The Nokia SNMP implementation lets an SNMP manager monitor the system and modify selected objects only. You can define and change one read-only community string and one read-write community string. You can set, add, and delete trap receivers and enable or disable various traps. You can also enter the location and contact strings for the system.

For more detailed information about the MIBs that the Nokia implementation supports, see the online Voyager documentation that comes with the system. You can also download a pdf version of the online documentation from the Nokia support site. at https://support.nokia.com. To view detailed information about each supported MIB, go to the /etc/snmp/mibs directory.

The Nokia implementation also supports the User-based Security model (USM) portion of SNMPv3.

SNMP Command Set

Use the following commands for configuring SNMP parameters.

```
set snmp
        daemon <on | off>
        snmp smp-version <v1/v2/v3 | v3-Only>
        trapreceiver ip_address community string version <v1 | v2>
        trapreceiver ip_address version <v1 | v2>
        trapPduAgent ip_address
        location string
        contact string
```

Use the following commands to configure SNMP traps.

```
set snmp traps
        coldstart status <on | off>
        link-up-down status <on | off>
        authorization status <on | off>
        vrrp-newmaster status <on | off>
        vrrp-authfail status <on | off>
        sys-config-change status <on | off>
        sys-config-filechange status <on | off>
        sys-config-savechange status <on | off>
        sys-lowdiskspace status <on | off>
        sys-nodiskspace status <on | off>
        sys-diskfailure status <on | off>
        sys-diskmirr-create status <on | off>
        sys-diskmirr-delete status <on | off>
        sys-diskmirr-syncfail status <on | off>
        sys-diskmirr-syncsuccess status <on | off>
        cluster-member-join status <on | off>
        cluster-member-left status <on | off>
        cluster-new-master status <on | off>
        cluster-member-reject status <on | off>
        cluster-protocol-interface-change status <on | off>
        sys-fan-failure status <on | off>
        sys-powersupply-failure status <on | off>
        sys-overtemperature status <on | off>
```

Use the following commands to configure other SNMP parameters.

```
add snmp
        address ipaddress
        community string read-only
        community string read-write
```

> **Note**
> The default community string is public.

```
        trapreceiver ip_addr community string version <v1 | v2>

delete snmp
        address ipaddress
        community string read-only
        community string read-write
        trapreceiver ip_address
```

For more detailed information about how to enable SNMP and configure basic settings, see "Enabling/Disabling and Setting SNMP" on page 234.

For more detailed information about SNMP traps, see "Enabling and Disabling SNMP Traps" on page 237.

For more detailed information about SNMPv3 and USM Users, see "Managing SNMP Users" on page 241.

Enabling/Disabling and Setting SNMP

Use the following commands to enable or disable SNMP and to set and change such parameters as the community strings, the Trap Receiver and PDU Agent address.

```
set snmp
        daemon <on | off>
```

Caution
If you run the Check Point and IPSO SNMP daemons simultaneously, you must start the Check Point SNMP daemon after you start VPN-1/FireWall-1 NG. If you start the Check Point daemon before you start VPN-1/FireWall-1 NG, the IPSO daemon does not start.

```
        snmp smp-version <v1/v2/v3 | v3-Only>
        trapreceiver ip_address community string version <v1 | v2>
        trapreceiver ip_address version <v1 | v2>
        trapPduAgent ip_address
        location string
        contact string

add snmp
        address ip_address
        community string read-only
        community string read-write
        trapreceiver ip_addr community string version <v1 | v2>

delete snmp
        address ip_address
        community string read-only
        community string read-write
        trapreceiver ip_address
```

Note
Use the set commands to configure initial settings and use the add commands to configure community strings and additional trap receivers.

Arguments

`daemon <on | off>`	Specifies whether to enable or disable SNMP. **Default:** on
`snmp-version <v1/v2/v3| v3-Only>`	Specifies which version of SNMP to implement. Selecing access limits community access. Only requests from users with enabled v3 access are allowed. All other requests are rejected. v1/v2/v3 allows the use of community names. **Default:** v1/v2/v3
`snmp address` *`ip_address`*	Specifies a IP address on which the agent responds to requests. The default is for the protocol to respond to requests from all interfaces. If you set a specific address, and want to revert to the default, use the delete snmp ip_address command.
`snmp community` *`string`* `read-only`	Sets a read-only community string. Use alphanumeric characters with no spaces, the hypen symbol and the underscore symobl only. If you delete the read-only community strings, SNMP GETS are not possible unless a read-write community string is configured that equals the input read community string.
`snmp community` *`string`* `read-write`	Sets a read-write community string. Use alphanumeric characters with no spaces, the hyphen symbol and the underscore symbol only. If you disable the SNMP community read-write string, SNMP SETS are not possible.

`trapreceiver` *`ip_address`* `community` *`string`* `version <v1 | v2>`	Specifies the IP address of a new receiver to accept traps from this system and the receiver's corresponding string. For the string, use alphanumeric characters with no spaces. You can add multiple receivers. The Nokia implementation supports using version 1 or version 2. The string for a receiver has no relationship with the read-only or read-write community strings. If you do not configure a string for the receiver, the string defaults to public **Default:** v1
`trapPduAgent` *`ip_address`*	Specifies the address used as the agent address in the protocol data unit of traps sent. This IP address must belong to a configured interface Beginning with IPSO 3.7, if you do not configure a Trap PDU Agent address, the system identifies the PDU Trap Agent address as 0.0.0.0 in SNMP traps. This change is in accordance with RFC 2089. For all previous releases of IPSO, the default was to use the IP address of the first valid interface.
`location` *`string`*	Specifies a string that contains the location for the system. The maximum length for the string is 128 characters including letters, numbers, spaces, special characters. For example: *`Bldg 1, Floor 3, WAN Lab, Fast Networks, Speedy, CA`*

`contact` *`string`*	Specifies a string that contains the contact information for the device. The maximum length for the string is 128 characters including letters, numbers, spaces, special characters. For example: *`John Doe, Network Administrator, (111) 222-3333`*

Enabling and Disabling SNMP Traps

Use the following command to enable or disable individual SNMP Traps.

Note
Only the cold start and authorization traps are enabled by default. You must enable all other traps.

```
set snmp traps
        coldstart status <on | off>
        link-up-down status <on | off>
        authorization status <on | off>
        vrrp-newmaster status <on | off>
        vrrp-authfail status <on | off>
        sys-config-change status <on | off>
        sys-config-filechange status <on | off>
        sys-config-savechange status <on | off>
        sys-lowdiskspace status <on | off>
        sys-nodiskspace status <on | off>
        sys-diskfailure status <on | off>
        sys-diskmirr-create status <on | off>
        sys-diskmirr-delete status <on | off>
        sys-diskmirr-syncfail status <on | off>
        sys-diskmirr-syncsuccess status <on | off>
        cluster-member-join status <on | off>
        cluster-member-left status <on | off>
        cluster-new-master status <on | off>
        cluster-member-reject status <on | off>
        cluster-protocol-interface-change status <on | off>
        sys-fan-failure status <on | off>
        sys-powersupply-failure status <on | off>
        sys-overtemperature status <on | off>
```

Arguments

`status <on | off>`	Specifies whether to enable or disable the specified trap.
`coldstart`	coldStart trap signifies that the SNMPv2 entity, acting in an agent role, is reinitializing itself and that its configuration might have been altered. The coldstart trap is enabled by defaut

`link-up-down`	ifLinkUpDown trap is sent when one of the links, which is administratively up, either has come up or been lost. The linkUpDown trap is enabled by default.
`authorization`	authenticationFailure trap sends notification that the SNMP message received from the sending entity is not properly authenticated.
`vrrp-newmaster`	vrrpTrapNew Master sends notification of a new VRRP master router.
`vrrp-authfail`	vrrpTrapAuthFailure sends notification of a VRRP authentication failure.
`sys-config-change`	systemTrapConfigurationChange is sent when a change is made to the running system configuration
`sys-config-filechange`	systemTrapConfigurationFileChange is sent when a change is made to system configuration files
`sys-config-savechange`	systemTrapConfigurationSaveChange is sent when a change is made to the running system configuration and saved to the database.
`sys-lowdiskspace`	systemTrapLowDiskSpace sent when the disk utilization, as seen by a non-superuser, in any of the local file systems exceeds 80%. The trap is initially sent within the scan interval (currently 30 seconds), and subsequently, at preset intervals of 15 minutes, until the disk utilization falls below 80%.

`sys-nodiskspace`	systemTrapNoDiskSpace is sent when the disk utilization, as seen by a non-superuser, in any of the local file systems exceeds 98%. The trap is initially sent within the scan interval (currently 30 seconds), and subsequently, at preset intervals of 15 minutes, until the disk utilization falls below 98%.
`sys-diskfailure`	systemTrapDiskFailure is sent when a particular disk drive fails, that is, there is no response from the disk for read/write operations. This trap applies on the IP530 and the IP740.
`sys-diskmirr-create`	systemTrapDiskMirrorSetCreate is sent when a particular mirror set has been created on the system.
`sys-diskmirr-delete`	systemTrapDiskMirrorSetDelete is sent when a particular mirror set has been deleted from the system.
`sys-diskmirr-syncfail`	systemTrapDiskMirrorSyncFailure is sent when a particular mirror set fails during syncing.
`sys-diskmirr-syncsuccess`	systemTrapDiskMirrorSyncSuccess is sent when a particular mirror set has been successfully synced.
`cluster-member-join`	ipsoLBClusterMemberJoin trap is sent when a member node joins the cluster by the master.
`cluster-member-left`	ipsoLBClusterMemberLeft trap is sent when a member node leaves the cluster by the master.

`cluster-new-master`	ipsoLBClusterNewMember trap is sent when a cluster is formed and a new master is elected.
`cluster-member-reject`	ipsoLBJoinReject trap is sent when a member's request to join a cluster is rejected.
`cluster-protocol-interface-change`	clusterProtocolInterfaceChange trap is sent when a failover occurs from the primary cluster to the secondary cluster network.
`sys-fan-failure`	systemFanFailure trap is sent when a fan fails. This trap includes the fan index and is supported only on the IP530 and IP740 platforms.
`sys-powersupply-failure`	systemPowerSupplyFailure trap is sent when a power supply for the system fails. This trap includes the power supply index and is supported only on the IP530 and IP740 platforms.
`sys-overtemperature`	systemOverTemperature trap is sent when a power supply failure occurs because of high temperature. This trap is followed by a power supply failure that specifies the power supply index that failed. This trap is supported only on the IP530 and IP740 platforms.

Managing SNMP Users

Use the following commands to add users who are authorized to use SNMPv3.

```
add snmp usm user username
        seclvl <authPriv | authNoPriv | authPrivReq>
           authpassphrase authphrase privpassphrase privacyphrase
```

Use the following command to change a user security-level setting or pass phrases.

```
set snmp usm user username
        seclvl <authPriv | authNoPriv | authPrivReq>
           authpassphrase authphrase privpassphrase privacyphrase
```

Use the following command to delete an existing SNMP user.

```
delete snmp usm user username
```

Use the following command to view existing SNMP users.

```
show snmp usm user username
```

```
show snmp users
```

Arguments

username	Range: 1 to 31 alphanumeric characters with no spaces, backslash, or colon characters.
seclvl <authPriv | authNoPriv | authPrivReq>	Security Level. Select from the following: • authNoPriv—User has authentication and privacy pass phrases and can connect with or without privacy encryption. • authPriv—User has only an authentication pass phrase and can connect only without privacy encryption. • authPriv—User must use authentication and privacy encryption pass phrases to connect.
authpassphrase	Range: 8-128 characters.
privpassphrase	Range: 8-128 characters.

Show SNMP Implementation and Trap Commands

```
show snmp
        daemon
        community
        trapreceiver
        traps
        snmp trapPduAgent
        snmp location
        snmp contact
```

SNMP Error Messages

This section lists and explains certain common error status values that can appear in SNMP messages. Within the protocol-data unit (PDU), the third field can include an error-status integer that refers to a specific problem. The integer zero (0) means that no errors were detected. When the error-field is anything other than 0, the next field, includes an error-index value that identifies the variable, or object, in the variable-bindings list that caused the error.

See the table below for the error status codes and their corresponding meanings.

Error Status Code	Meaning
0	noError
1	tooBig
2	NoSuchName
3	BadValue
4	ReadOnly
5	genError
6	noAccess
7	wrongType

Error Status Code	Meaning
8	wrongLength
9	wrongEncoding
10	wrongValue
11	noCreation
12	inconsistentValue
13	resourceUnavailable
14	commitFailed
15	undoFailed
16	authorizationError
17	notWritable
18	inconsistentName

Note
You do not necessarily see the codes. The SNMP manager or utility interprets the codes and displays and logs the appropriate message.

The subsequent, or fourth field, contains the error-index when the error-status field is nonzero, that is, when the error-status field returns a value other than zero, which indicates that an error occurred. The error-index value identifies the variable, or object, in the variable-bindings list that caused the error. The first variable in the list has index 1, the second has index 2, and so on.

The next, or fifth field, is the variable-bindings field. It consists of a sequence of pairs; the first is the identifier. The second element is one the following five: value,

unSpecified, noSuchOjbect, noSuchInstance, and EndofMibView. The table below describes each element.

Variable-Bindings element	Description
value	the value associated with each object instance; specified in a PDU request
unSpecified	a NULL value is used in retrieval requests
noSuchObject	indicates that the agent does not implement the object referred to by this object identifier
noSuchInstance	indicates that this object does not exist for this operation
endOfMIBView	indicates an attempt to reference an object identifier that is beyond the end of the MIB at the agent

GetRequest

The following are possible value field sets in the response PDU or error-status messages when preforming a *GetRequest*

noSuchObject	If a variable does not have an *OBJECT IDENTIFIER* prefix that exactly matches the prefix of any variable accessible by this request, then its value field is set to *noSuchObject*.
noSuchInstance	If the variable's name does not exactly match the name of a variable, then its value field is set to *noSuchInstance*.
genErr	If the processing of a variable fails for any other reason, the responding entity returns *genErr* and a value in the error-index field that is the index of the problem object in the variable-bindings field.
tooBig	If the size of the message that encapsulates the generated response PDU exceeds a local limitation or the maximum message size of the request's source party, then the response PDU is discarded and a new response PDU is constructed. The new response PDU has an error-status of *tooBig*, an *error-index* of zero, and an empty *variable-bindings* field.

GetNextRequest

The only values that can be returned in as the second element in the variable-bindings field to a GetNextRequest when an error-status code occurs are unSpecified or endOfMibView.

GetBulkRequest

The GetBulkRequest minimizes the number of protocol exchanges by letting an SNMPv2 manager request that the response be as large as possible given the constraints on the message size.

The GetBulkRequest PDU has two fields that do not appear in the other PDUs: non-repeaters and max-repetitions. The non-repeaters field specifies the number of variables in the variable-bindings list for which a single-lexicographic successor is to

be returned. The max-repetitions field specifies the number of lexicographic successors to be returned for the remaining variables in the variable-bindings list.

If at any point in the process, a lexicographic successor does not exist, the endofMibView value is returned with the name of the last lexicographic successor, or, if there were no successors, the name of the variable in the request.

If the processing of a variable name fails for any reason other than endofMibView, no values are returned. Instead, the responding entity returns a response PDU with an error-status of genErr and a value in the error-index field that the is the index of the problem object in the variable-bindings field.

7 IPv6 Commands

Use the commands in this chapter to configure most IPv6 settings for your system. To configure IPsec for IPv6, see “IPsec Commands (IPSO Implementation)” on page 291. To configure traffic management for IPv6, see “Clustering Administration” on page 224.

Configuration Summary

Use the following command to show a summary of IPv6 configuration on your system:

```
show ipv6 config
```

Interface Commands

Use the following commands to associate an IPv6 address with a logical interface, anycast address, or IPv6 address family:

```
add interface if_name
        ipv6prefix ip6_address/mask
        anycast ip6_address
        family inet6
```

Use the following commands to disassociate an IPv6 address from a logical interface, anycast address, or IPv6 address family:

```
delete interface if_name
        ipv6prefix ip6_address/mask
        anycast ip6_address
        family inet6
```

Arguments

`interface` *`if_name`*	Specifies the name of an existing logical interface. **Note** You cannot disable an IPv6 inteface configured for a virtual router when the router is in the master state. If you try to disable the interface, when the router is in the master state, the console displays an error message. To disable the IPv6 interface, you must first delete the interface as a VRRP virtual address. You can, however, disable an IPv6 interface enabled on a virtual router when the router is in a backup state.
`ipv6prefix` *`ip6_address/mask`*	Specifies the IPv6 address and mask length. The mask length range is <8–126>. • Format: IPv6 Prefix/<8–126> • Example: 1000:50:32::3/64
`anycast` *`ip6_address`*	Specifies an anycast address. When you assign an IPv6 anycast address to multiple interfaces (typically on different systems), packets sent to an anycast address are routed to the nearest interface that matches the address of the packet.
`family inet6`	Add an IPv6 address family to the specified logical interface.

Use the following commands to view information about IPv6 interfaces configured on your system.

```
show ipv6
        interfaces
        interface if_name

show interface if_name ipv6prefix
```

Arguments

`interfaces`	Displays summary information about all configured IPv6 interfaces.
`interface if_name`	Displays information about a specified logical interface.
`interface if_name ipv6prefix`	Displays configured IPv6 prefixes for the specified logical interface.

Neighbor Discovery Protocol

The Neighbor Discovery Protocol (NDP) allows you to map an IPv6 address to a physical machine address recognized in the local network.

Use the following command to add a new static Neighbor Discovery entry:

```
add neighbor-entry address ip6_address macaddress mac_address
```

Use the following command to remove a static Neighbor Discovery entry:

```
delete neighbor-entry address ip6_address
```

Arguments

`address` *`ip6_address`*	Specifies the IPv6 address of the static NDP entry to add or delete. • Format: IPv6 address • Example: 1000:50:32::2
`macaddress` *`mac_address`*	Specifies the MAC address for the associated IPv6 interface address. • Format: hexadecimal digits • Example: 00:a0:8e:86:73:60

Use the following commands to configure global NDP properties:

```
set neighbor
        duplicate-detection <1-100>
        multicast-limit <1-100>
        queue-limit <1-3>
        unicast-limit <1-100>
```

Arguments

`duplicate-detection <1-100>`	Specifies the number of times to retry Duplicate Address Detection NDP requests. • Default: 3
`multicast-limit <1-100>`	Specifies the number of times to retry Multicast NDP requests. • Default: 3
`queue-limit <1-3>`	Specifies the maximum number of output packets to be queued while resolving link-layer destination address. **Default:** 3

`unicast-limit <1-100>`	Specifies the number of times to retry Unicast NDP requests. **Default:** 3

Use the following commands to view NDP configuration details:

```
show neighbor
        dynamic-table
        interface-table
        parameters
        static-table
        table
```

Arguments

`dynamic-table`	Displays the dynamically learned neighbor IPv6 addresses and thier respective MAC addresses.
`interface-table`	Displays the IPv6 addresses and MAC addresses of currently configured interfaces listed in the neighbor table.
`parameters`	Displays neighbor table parameters, including each configurable field and its associated value.
`static-table`	Displays neighbor table static entries.
`table`	Displays the entire neighbor table including static entries, interface entries, and dynamic entries.

Tunnels

Tunnels are point-to-point links that transport packets from a source interface to a destination interface.

Use the following commands to create tunnels by using a specified encapsulation scheme. For GRE and DVMRP tunnels, add an interface with this encapsulation first and then set the tunnel endpoints with a separate `set` command. To encapsulate IPv4 packets in IPv6 tunnels or IPv4 packets in IPv6 tunnels, use the `add` command to select the interface and specify the tunnel endpoint addresses. You can specify other optional arguments depending on the encapsulation scheme you select.

```
add interface phys_if_name encapsulation
        dvmrp
        gre
        v6inv4 address ip_address remote ip_address [local-link-local
           linklocal_address] [remote-link-local linklocal_address] [ttl
           <1-255>]
        v4inv6 address ip6_address remote ip6_address
```

Arguments

`interface` *`phy_if_name`*	Specifies the physical tunnel interface name. The value must be a tunnel that exists on the system. • Example: tun0
`gre`	Specifies a GRE tunnel.
`dvmrp`	Specifies a DVMRP tunnel.
`v6inv4`	Specifies a tunnel that encapsulates IPv6 packets in IPv4 packets. This argument allows you to connect IPv6 enabled interfaces (typically on different systems) over existing IPv4 connections.

`v4inv6`	Specifies a tunnel that encapsulates IPv4 packets in IPv6 packets. This argument allows you to connect IPv4 enabled systems over existing IPv6 connections.
`address <`*`ipv4_address`* `|` *`ip6_address`*`>`	Specifies the interface address of the local tunnel endpoint. If you use IPv6 in IPv4 encapsulation, the local address is in IPv4 address format. For IPv4 in IPv6 encapsulation, use a valid IPv6 address format: • IPv4 example: 192.168.50.5 • IPv6 example: 2222::1:2:3
`destination <`*`ipv4_address`* `|` *`ip6_address`*`>`	Specifies the interface address of the remote tunnel endpoint. If you use IPv6 in IPv4 encapsulation, the destination address is in IPv4 address format. For IPv4 in IPv6 encapsulation, use a valid IPv6 address format: • IPv4 example: 192.168.80.8 • IPv6 example: 2222::4:5:6
`local-link-local` *`linklocal_address`*	Specifies the link-local address of the local interface to which the local end of the tunnel is bound. This argument is optional. If you specify an address, it should be unique. In other words, it should not be a link-local address that already exists on your system. Example: FE80::32

`remote-link-local` *linklocal_address*	Specifies the link-local address of the interface on the remote system to which the remote end of the tunnel is bound. This argument is optional. If you specify an address, it should be unique. In other words, it should not be a link-local address that already exists on the remote system. Example: FE80::52
`ttl <1-225>`	Specifies the time to live of packets sent on the tunnel. This argument is optional.

Use the following commands to configure properties for existing tunnels:

```
set interface if_name
        interface-binding <on | off>
        local-endpoint <ip_address | ip6_address> <enable | disable>
        address <ip_address | ip6_address> destination
           <ip_address | ip6_address> remote-endpoint
           <ip_address | ip6_address>
```

Arguments

`interface` *if_name*	Specifies the logical tunnel interface name. The value must be a tunnel that exists on the system: • Example: tun0c3
`interface-binding <on | off>`	Specifies whether to bind a tunnel to an outgoing interface. • `on: Bind the tunnel to the interface.` • `off:` Do not bind the tunnel to the interface. **Default:** on

`local-endpoint <`*`ip_address`* `| `*`ip6_address`*`> <enable | disable>`	Specifies the IPv6 or IPv4 address of the local interface to which the local end of the tunnel is bound. Disabling the tunnel will not delete the configured address information.
`address <`*`ipv4_address`* `| `*`ip6_address`*`>`	Specifies the address of the local tunnel endpoint. If the tunnel uses IPv6 in IPv4 encapsulation, the local address will be in IPv4 address format. For IPv4 in IPv6 encapsulation, use a valid IPv6 address format: • IPv4 example: 192.168.50.5 • IPv6 example: 2222::1:2:3
`destination <`*`ipv4_address`* `| `*`ip6_address`*`>`	Specifies the address of the remote tunnel endpoint. If the tunnel uses IPv6 in IPv4 encapsulation, the destination address will be in IPv4 address format. For IPv4 in IPv6 encapsulation, use a valid IPv6 address format: • IPv4 example: 192.168.80.8 • IPv6 example: 2222::4:5:6
`remote-endpoint <`*`ipv4_address`* `| `*`ip6_address`*`>`	Specifies the IPv6 or IPv4 address of the interface on the remote system to which the remote end of the tunnel is bound.

Use the following command to delete a specified logical tunnel:

```
delete interface if_name
```

Arguments

`interface` *`if_name`*	Specifies the logical interface name of the tunnel to delete: • Example: tun0c3

Use the following command to view summary information about the IPv6 tunnels configured on your system:

```
show ipv6 tunnels
```

IPv6 to IPv4

Use the commands in this section to configure an IPv6 interface attached to an IPv4 network that does not have IPv6 native support. This feature allows you to connect IPv6 domains through IPv4 clouds without explicit tunnels.

Use the following commands to create and configure an IPv6 to IPv4 interface, or to delete existing IPv6 to IPv4 settings. The time to live (TTL) argument is optional.

```
set ipv6toipv4
        active on address ip_address enable [ttl <1–255>]
        disable
```

Use the following command to activate or deactivate an existing IPv6 to IPv4 interface:

```
set ipv6toipv4 active <on | off>
```

Arguments

`active on address` *`ip_address`*	Activates the IPv6 to IPv4 feature. You must specify the local IPv4 address to activate this feature if the interface association does not already exist or if you disable it.

`ttl <1-255>`	Specifies the time to live (TTL) of packets sent on the tunnel. This argument is optional. Default: 255
`disable`	Deletes the settings.
`<on | off>`	Activates or deactivates existing IPv6 to IPv4 settings. If you use `active off`, you do not lose the current settings.

Use the following command to view the IPv6 to IPv4 configuration on your system:

```
show ipv6toipv4
```

IPv6 Over IPv4

Use the following commands to create and enable or disable an IPv6 interface attached to an IPv4 network that does not have IPv6 native support. This feature allows you to transmit IPv6 traffic over IPv4 domains without explicit tunnels.

Use the following commands to create and configure IPv6 to IPv4 features, or to delete IPv6 to IPv4 settings. The time to live (TTL) argument is optional.

```
set ipv6overipv4
        active on address ip_address enable [ttl <1-255>]
        disable
```

Use the following command to activate or deactivate the IPv6 to IPv4 settings for an interface.

```
set ipv6overIPv4 active <on | off>
```

Arguments

`active on address` *`ip_address`*	Activates the IPv6 over IPv4 feature. You must specify the local IPv4 address to activate this feature if the interface association does not already exist or if you disable it.
`ttl <1-255>`	Specifies the time to live (TTL) of packets sent on the tunnel. This argument is optional. Default: 255
`disable`	Deletes the settings.
`<on | off>`	Activates or deactivates existing IPv6 over IPv4 settings. If you use `active off`, you will not lose the current settings.

Use the following command to view the IPv6 to IPv4 configuration on your system:

```
show ipv6overipv4
```

IPv6 Routing Configuration

Use the commands in this section to configure IPv6 routing on your system.

RIPng

Use this group of commands to set and view parameters for RIP next generation (RIPng).

Note
IPSO does not have CLI commands for route filtering and redistribution. You must configure inbound routing policies and redistribution of routes

through Voyager. You can configure route maps and route aggregation using CLI commands. Route map configuration done through the CLI takes precedence over route filtering and redistribution configured in Voyager. For example if RIP uses route maps for inbound filtering, anything configured on the Voyager page for inbound route filters for RIP is ignored. You can still use Voyager to configure route redistribution into RIP.

Interfaces

Use the following commands to configure RIPng properties for specific interfaces:

```
set ipv6 ripng interface if_name
        <on | off >
        metric <0-16>
        metric default
```

Arguments

`<on | off>`	Specifies whether to run RIPng on the specified interface.
`metric <0-16>`	Specifies the RIP metric added to routes that use the specified interface.
`metric default`	Specifies a value of 0.

Show Commands

Use the following commands to monitor and troubleshoot RIPng:

```
show ipv6 ripng
        interfaces
        interface if_name
        packets
        errors
        neighbors
        summary
```

Route Aggregation

Use the following group of commands to aggregate numerous specific routes into one route. Route aggregation potentially reduces the number of routes that given protocol advertises.

Only the receiver, and not the originator, of an aggregate route uses it for forwarding packets. The originator of the aggregate route uses individual component routes to determine reachability. A router that receives a packet that does not match one of the component routes of the aggregate responds with an Internet Control Message Protocol (ICMP) network unreachable message. This message prevents packets or unknown component routes from following a default route to another network where they would be continually forwarded back to the border router until their time to live (TTL) expires.

Create an aggregate route by first specifying the network address and mask length. Next, provide a set of contributing routes. To define a contributing route, specify a source (static route, interface route, or routing protocol) and a route filter (an IPv6 prefix). An aggregate route can have many contributing routes, but at least one of the routes must be present to generate an aggregate. The `off` argument deactivates a specified IPv6 aggregate route.

```
set ipv6 aggregate ip6_prefix
        off
        contributing-protocol
           <all | direct | static | aggregate | ripng> off
        contributing-protocol
           <all | direct | static | aggregate | ripng>
           contributing-route <all | ip6_prefix> <on | off>
```

Arguments

ip6_prefix	Specifies the IPv6 address and mask length of the new aggregate route and the contributing protocol or interface route: • Example: 1000:50:32::/64
off	Deactivates the specified aggregate route.
contributing-protocol <all | direct | static | aggregate | ripng> off	Specifies the contributing route source type or protocol to turn off: • all — Use all routes. • direct — Use only direct routes. • static — Use only static routes. • aggregate — Use only aggregate routes. • ripng — Use only routes that use ripng protocol.
contributing-route <all | *ip6_prefix*> <on | off>	Specifies the contributing route to turn on or off for the specific contributing protocol. You must use this argument after specifying a contributing protocol. • all — Contribute all the routes for a specific source type or protocol. • ip6_prefix — Contribute a specific route.

Static Routes

Static routes cause packets moving between a source and a destination to take a specified next hop. Static routes allow you to add routes to destinations that are not described by dynamic routing protocols. A static route can also be useful in providing a default route.

Use the following group of commands to configure specific static routes:

```
set ipv6 static-route
        ip6_prefix nexthop gateway ip6_address priority <1-8> <on | off>
        default nexthop gateway ip6_address priority < 1-8> <on | off>
        ip6_prefix nexthop gateway ip6_address interface if_name priority
           <1-8> <on | off>
        default nexthop gateway ip6_address interface if_name <on | off>
        ip6_prefix nexthop reject
        default nexthop reject
        ip6_prefix nexthop blackhole
        default nexthop blackhole
        ip6_prefix off
        default off
```

Arguments

`static-route` *`ip6_prefix`*	Specifies the IPv6 prefix and mask length of the static route. Use the `off` argument to disable the specified route. Use the other arguments to configure properties of the specified route and enable or disable them. Address example: 1000:50.32::/64
`static-route default`	Specifies the default static route. Use the `off` argument to disable the default route. Use the other arguments to configure default route properties and enable or disable them.

`nexthop gateway` *`ip6_address`* `<on | off>`	Specifies the gateway address and enables or disables it for the IP address configured as the endpoint of the static route from your system. Disabling the gateway address does not delete the route itself. Address example: 1000:50:32::1
`nexthop gateway` *`ip6_address`* `interface` *`if_name`* `<on | off>`	Specifies the gateway address for a specified interface and enables or disables it for the IP address configured as the endpoint of the static route from your system. Disabling the gateway address does not delete the route itself. Address example: 1000:50:32::1 Interface example: eth-s1p1c0
`priority <1-8>`	Specifies the order in which next hops are selected. The lower the value the more preferred the link. The next preferred value is selected as the next hop only when an interface fails. A non-reachable link is not selected as the next hop. NOTE: The priority option also supports equal-cost multipath routing. For each priority value, you can configure as many as eight gateway addresses. The nexthop gate address for each packet to the destination is selected based on the nexthop algorithm that is configured.
`ip6_prefix` `nexthop reject`	Specifies for packets to be dropped rather than forwarded and for unreachable messages to be sent to the packet originators. Specifying this option causes this route to be installed as a reject route.

ip6_prefix nexthop backhole	Specifies for packets to be dropped rather than forwarded but does **not** specify for unreachable messages to be sent to the packet originator.
ip6_prefix off	Deletes the specified static route and deletes any next hops associated with the route.
default off	Deletes the default route and deletes any next hops associated with the route.

ICMP Router Discovery

Use this group of commands to set and view parameters for the ICMP router discovery protocol.

Interfaces

Use the following commands to configure router discovery properties for specific interfaces:

```
set ipv6 rdisc6 interface if_name
        <on | off>
        min-adv-interval <3-1800>
        min-adv-interval default
        max-adv-interval <4-1800>
        max-adv-interval default
        hop-limit <0–255>
        hop-limit default
        managed-config <on |off>
        other-config <on | off>
        reachable-time <0–3600000>
        reachable-time default
        retransmit-timer integer
        retransmit-timer default
        router-lifetime integer
        router-lifetime default
        send-mtu <on | off>
```

Use the following commands only if the mask length is *not* greater than 64:

```
set ipv6 rdisc6 interface if_name
        address ip6_address autonomous <on | off>
        address ip6_address on-link <on | off>
        address ip6_address prefix-pref-lifetime integer
        address ip6_address prefix-pref-lifetime default
        address ip6_address prefix-valid-lifetime integer
        address ip6_address prefix-valid-lifetime default
```

Arguments

<on | off>	Specifies whether to run ICMPv6 router discovery on a specified interface.

`min-adv-interval <3-1800>`	Specifies the minimum time (in seconds) allowed between sending unsolicited broadcast or multicast ICMPv6 router advertisements on the interface. **Note** Beginning with IPSO 3.8.1 and as part of the new support of VRRP for IPv6 interfaces, only the router in a VRRP master state sends router discovery advertisements, and the advertisemetns are sent with the virtual IP address as the source address and the virtual MAC address as the MAC address. Routers in a VRRP bacukp state do not send router discovery advertisements. For more information about how to configure VRRP for IPv6 interfaces, see "VRRP for IPv6" on page 272.
`min-adv-interval default`	Specifies a value of 450 seconds.
`max-adv-interval <4-1800>`	Specifies the maximum time (in seconds) allowed between sending unsolicited broadcast or multicast ICMPv6 router advertisements on the interface.
`max-adv-interval default`	Specifies a value of 600 seconds.
`hop-limit <0-255>`	Specifies the value placed in the `Cur Hop Limit` field in the router advertisement packet. Systems use this value in the `Hop Count` field of the IP header for outgoing IP packets. This value should be set to the current diameter of the Internet. The value zero (0) means unspecified (by this router).
`hop-limit default`	Specifies a value of 64.

`managed-config <on | off>`	Specifies whether to perform stateful autoconfiguration to obtain addresses. The `Managed Config` flag is placed in the `Managed Address Configuration Flag` field in the router advertisement packet. When this flag is set to `yes`, hosts perform stateful autoconfiguration to obtain addresses. **Default:** off
`other-config <on | off>`	Specifies whether to perform stateful autoconfiguration to obtain information other than addresses.The `Other Config` flag is placed in the `Other Stateful Configuration Flag` field in the router advertisement packet. When this flag is set to `yes`, hosts perform stateful autoconfiguration to obtain additional information (excluding addresses). **Default:** off
`reachable-time <0-3600000>`	Specifies the time a node assumes a neighbor is reachable after having received a reachability confirmation.The reachable time is placed in the `Reachable Time` field in the router advertisement packet. This value is used by the Neighbor Unreachability Detection. The value zero (0) means unspecified (by this router).
`reachable-time default`	Specifies a value of zero (0).

`retransmit-timer` *integer*	Specifies the time between retransmitted Neighbor Solicitation messages if the system does not receive a response.The retransmission timer is placed in the `Retrans Timer` field in the router advertisement packet. Address resolution and Neighbor Unreachability Detection uses this value. The value zero (0) means unspecified (by this router).
`retransmit-timer default`	Specifies a value of zero (0).
`router-lifetime` *integer*	Specifies the value (in seconds) placed in the `Router Lifetime` field of the router advertisements packet. A value of zero (0) indicates that the router is not to be used as a default router.
`router-lifetime default`	Specifies a value of 1800 seconds.
`send-mtu <on | off>`	Specifies whether the router advertisement packet includes MTU options. **Default:** off
`autonomous <on | off>`	Specifies whether this prefix can be used for autonomous address configuration. **Default:** on
`on-link <on | off>`	Specifies whether this prefix can be used for on-link determination.

`prefix-valid-lifetime` *`integer`*	Specifies the length of time in seconds (relative to the time the packet is sent) that the prefix is valid for the purpose of on-link determination. This value is placed in the `Valid Lifetime` field in the `Prefix Information` option. The designated value of all 1s (0xffffffff) represents infinity.
`prefix-valid-lifetime default`	Specifies a value of 2592000 seconds (30 days).
`prefix-pref-lifetime` *`integer`*	Specifies the length of time in seconds (relative to the time the packet is sent) that addresses generated from the prefix through stateless address autoconfiguration remain preferred.This value is placed in the `Preferred Lifetime` field in the `Prefix Information` option. The designated value of all 1s (0xffffffff) represents infinity, which means that the node can use the prefix in existing connections, but it is not valid for new connections.
`prefix-pref-lifetime default`	Specifies a value of 604800 seconds (7 days).

Show Commands

Use the following commands to monitor and troubleshoot your ICMP router discovery implementation:

```
show ipv6 rdisc6
        interfaces
        interface if_name
        stats
        summary
```

VRRP for IPv6

Beginning with IPSO 3.8.1, Nokia supports VRRP for IPv6 interfaces. Nokia supports two implementations:

- VRRP version 3
- Monitored circuit

The CLI commands for these implementations are explained in the following sections.

All implementations

The following command applies to all VRRP iimplementations for IPv6, that is, both VRRPv3 and Monitored Circuit

```
set ipv6 vrrp6
        monitor-firewall <on | off>
```

Arguments

monitor-firewall	Specifies to monitor the state of the firewall and respond appropriately. If a VRRP master detects that the firewall is not ready to handle traffic or is not functioning properly, the master fails over to a backup system. If all the firewalls on all the systems in the VRRP group are not ready to forward traffic, no traffic is forwarded. **Note** If firewall is not installed, this option does not affect the function of VRRP for IPv6. **Default:** on

VRRPv3

Use this group of commands to configure and set parameters for VRRP version 3 for IPv6 interfaces.

```
set ipv6 vrrp6 interface if_name
        off
        virtual-router vrid <1-255> address ip_adresss on
        virtual-router backup-vrid <1-255> address ip_address on
        vrid <1-255> off
        vrid <1-255> address ip_address <on | off>
        vrid <1-255> accept-mode <on | off>
        vrid <1-255> hello-interval <1-4095>
        vrid <1-255> hello interval default
        vrid <1-255> priority <1-254>
        vrid <1-255> preempt-mode <on | off>
        vrid <1-255> vmac-mode default-vmac
        vrid <1-255> vmac-mode extended-vmac
        vrid <1-255> vmac-mode interface-vmac
        vrid <1-255> vmac-mode static-vmac mac_address
```

Arguments

`off`	Specifies to disable VRRPv3 on the specified IPv6 interface
`virtual router vrid <1-255> address` *`ip_address`* `on`	Specifies the virtual router ID for the virtual router used to back up the IP addresses of the local interface. The vrid must be unique for all virtual routers running on the network of the interface. Specify an IPv6 address for the virtual router. The address configured must be a link-local address.
`virtual-router backup-vrid <1-255> address` *`ip_address`* `on`	Specifies the virtual router ID for the virtual router used the back up the IPv6 addresses of another system. The router you are backing up must also have this virtual router configured for its addresses. Specify an IPv6 address to assign to the virtual router used to back up the IP addresses of another system. The address configured must be a link-local address. Global addresses should belong to the interface's subnet and link-local addresses must belong to the fe80::/64 subnet.
`vrid <1-255> off`	Specifies to remove the specified virtual router.

`vrid <1-255> address` *`ip_address`* `<on | off>`	Specifies to add or remove the IP address for the specified local or backup virtual router. Unlike the VRRP for IPv4 implementation, where a virtual router used to back up the IP addresses of the the local interface automatically picks up those addresses, you must manually configure the IP addresses for the virtual router to back up. For both a local virtual router and a virtual router used to back up the IPv6 addresses of a remote system, when a new subnet is added to the interface, you must manually configure any IPv6 addresses you want to include in the virtual router.
`vrid _<1-255> accept-mode <on | off>`	Specifies for the virtual router in a master state to accept packets addressed to virtual IP addresses. You must enable this otion when deploying dynamic routing protocols. **Default:** off.
`vrid <1-255> hello-interval <1-4095>`	Specifies in centiseconds, that is, in one-hundredths of a second, the interval between VRRP advertisement transmissions. **Default:** 100
`vrid <1-255> hello-interval default`	Specifies to set the interval between VRRP advertisements to 100 centiseconds, that is, one second.
`vrid <1-255> priority <1-254>`	Specifies the priotiy of this physical router during contention for the IPv6 addresses of a failed router. The router with the highest priority becomes the new master when a failure occurs on the existing master. **Default:** 100

`vrid <1-255> preempt-mode <on | off>`	Specifies for this virtual router to become the new master router if has a higher priority than the current master router. Enter off to disable this feature. **Default:** on
`vrid <1-255> vmac-mode <default-vmac | extended-vmac | interface-vmac | static-vmac mac_address>`	Specifies how the virtual MAC address for the backup address is created. You must choose the same option on each physical router that participates on the virtual router. • default-vmac: Uses the VRRP protocol specification to set the virtual MAC address. If this option does not create a unique VMAC on the network, choose one of the other options. • extended-vmac: Causes the system to use several variable to calculate the VMAC address so that the virtual routers with the same ID do not use the same address. • interface-vmac: Specifies that the MAC address of the associated physical interface be used as the VMAC address. • static-vmac *mac_address*: Uses the specified MAC address **Default:** default-vmac

Monitored Circuit for IPv6 Interfaces

Use the following group of commands to configure and set parameters for monitored circuit for IPv6 interfaces

```
set ipv6 vrrp6 interface if_name
        off
        monitored-circuit vrid <1-255> address ip_address on
        vrid <1-255> off
        vrid <1-255> address ip_address <on | off>
        vrid <1-255> accept-mode <on | off>
        vrid <1-255> hello-interval <1-4095>
        vrid <1-255> hello interval default
        vrid <1-255> monitored-interface if_name priority-delta <1-254>
           <on | off>
        vrid <1-255> monitored-interface if_name off
        vrid <1-255> preempt-mode <on | off>
        vrid <1-255> auto-deactivation <on | off>
        vrid <1-255> vmac-mode default-vmac
        vrid <1-255> vmac-mode extended-vmac
        vrid <1-255> vmac-mode interface-vmac
        vrid <1-255> vmac-mode static-vmac mac_address
```

Arguments

`off`	Specifies to disable monitored circuit on the specified IPv6 interface
`monitored-circuit vrid <1-255> address ip_address <on | off>`	Specifies the ID for a virtual router with monitored circuit dependencies and anIP address for the virtual router. The address configured must be a link-local address. Global addresses should belong to the interface's subnet, and link-local addresses must belong to the fe80::/64 subnet.
`vrid <1-255> off`	Specifies to remove the specified virtual router with monitored circuit dependencies.
`vrid <1-255> address ip_address <on | off>`	Specifies to add or delete an IPv6 address for this virtual router.

`vrid <1-255> accept-mode <on | off>`	Specifies for the virtual router in a master state to accept packets addressed to virtual IP addresses. **Default:** off
`vrid <1-255> hello-interval <1-4095>`	Specifies in centiseconds, that is, in one-hundredths of a second, the interval between VRRP advertisement transmissions. **Default:** 100
`vrid <1-255> hello-interval default`	Specifies to set the interval between VRRP advertisements to 100 centiseconds, that is, one second.
`vrid <1-255> monitored-interface` *`if_name`* `priority delta <1-254> <on | off>`	Specifies an interface with a dependency on the virtual router and the priority delta associated with the interface you selected. When an interface goes down, the priority delta value for the that interface is subtracted from the base priority value of the virtual router, resulting in the effective priority value. This effective priority value of the virtual router is used to determine the election of the VRRP master router. You can also use this command to change the priority delta of an existing monitored interface.
`vrid <1-255> monitored-interface` *`if_name`* `off`	Specifies to remove the monitored interface associated with the specified virtual router.
`vrid <1-255> preempt-mode <on | off>`	Specifies for this virtual router to become the new master router if has a higher priority than the current master router. Enter off to disable this feature. **Default:** on

`vrid <1-255> autodeactivation default <on | off>`	Specifies to set the minimum value for the effective priority of the virtual router to zero (0). The default is off, which sets the lowest value for the effective priority of the virtual router to one (1). A VRRP virtual router with an effective priority of 0 does not bcomes the master even if there are not other VRRP routers with a higher priority for this virtual router. **Default:** off
`vrid <1-255> vmac-mode <default-vmac | extended-vmac | interface-vmac | static-vmac mac_address>`	Specifies how the virtual MAC address for the backup address is created. You must choose the same option on each physical router that participates on the virtual router. • default-vmac: Uses the VRRP protocol specification to set the virtual MAC address. If this option does not create a unique VMAC on the network, choose one of the other options. • extended-vmac: Causes the system to use several variable to calculate the VMAC address so that the virtual routers with the same ID do not use the same address. • interface-vmac: Specifies that the MAC address of the associated physical interface be used as the VMAC address. • static-vmac *mac_address*: Uses the specified MAC address **Default:** default-vmac

VRRP for IPv6 Show Commands

```
show ipv6 vrrp6
        interface if_name
        interfaces
        stats
        summary
```

Show Routing Summary Commands

Use the commands in this section to view summary information about routes on your system.

Use the following command for information about active, inactive, and all RIPng routes on your system:

```
show ipv6 route
        ripng
        inactive ripng
        all ripng
```

Use the following command to show information about active, inactive, and all aggregate routes on your system:

```
show ipv6 route
        aggregate
        inactive aggregate
        all aggregate
```

Use the following command to show additional information about routes on your system:

```
show ipv6 route
        all
        all direct
        all static
        direct
        inactive
        inactive direct
        inactive static
        static
        summary
        destination ipv6_address
        exact ipv6_prefix
        less-specific ipv6_prefix
        more-specific ipv6_prefix
```

Host Name Configuration

Use the following commands to add or delete logical IPv6 hosts on your system:

```
add ipv6host
        localhost
        name name ipv6 ip6_address
```

```
delete ipv6host
        localhost
        name name
```

Use the following command to change the IPv6 address associated with the specified host name:

```
set ipv6host name name ipv6 ip6_address
```

Arguments

`localhost`	Adds or deletes an IPv6 local host. The associated address is ::1.

`ipv6host name` *`name`*	Specifies a logical host name. Use alphanumeric characters, dashes (-), and periods (.); however, periods must be followed by a letter or a digit. The host name cannot end in a dash or a period.
`ipv6` *`ip6_address`*	Specifies the IPv6 address to associate with the host name: Example: 1000:50:32::2

Use the following command to view the logical hosts and the associated IPv6 addresses configured on your system:

```
show ipv6host names
```

Use the following command to view the IPv6 address associated with the specified static host:

```
show ipv6host name name ipv6
```

Network Access and Services

Use the following `set` commands to enable or disable network access to this system for FTP, TFTP, and TELNET sessions. Use the `show` commands to view the current status of network access to the system by using FTP, TFTP, and TELNET.

```
set
        ipv6ftpaccess <enable | disable>
        ipv6tftpaccess <enable | disable>
        ipv6telnetaccess <enable | disable>

show
        ipv6ftpaccess
        ipv6tftpaccess
        ipv6telnetaccess
```

8 Network Security and Access Commands

This chapter describes the commands that you use to manage the security and access features of your system. It also explains how to enable or disable a VPN accelerator card and display VPN acceleration information.

Network Access and Services

Use this group of commands to configure and view network access such as FTP, TFTP and telnet sessions.

Use the following commands to configure network access.

```
set net-access
        ftp <yes | no>
               port <1-65535>
        tftp <yes | no>
        telnet <yes | no>
        admin-net-login <yes | no>
        cli-http <yes | no>
        cli-https <yes | no>
        com2-login <yes | no>
        com3-login <yes | no>
        com4-login <yes | no>
```

Use the following command to configure a PCMCIA modem.

```
set modem com4 <country code>
```

Use the following commands to view network access configurations.

```
show
        net-access
        net-access ftp
        net-access tftp
        net-access telnet
        net-access admin-net-login
        net-access cli-http
        net-access cli-https
        net-access com2-login
        net-access com3-login
        net-access com4-login
```

Use the following commands to configure types of services.

```
set services
        echo <yes | no>
        discard <yes | no>
        chargen <yes | no>
        daytime <yes | no>
        time <yes | no>
```

Use the following commands to view service configurations.

```
show
        services
        services echo
        services discard
        services chargen
        services daytime
        services time
```

Arguments

`ftp <yes | no>`	Specifies FTP access to the platform. **Default:** no
`port <1-65535>`	Specifies a port on which the ftpd server listens. **Default:** 21
`tftp <yes | no>`	Specifies TFTP access to the platform. **Default:** no
`telnet <yes | no>`	Specifies telnet access to the platform. **Default:** no
`admin-net-login <yes | no>`	Specifies “admin” login for telnet access to the platform. This will not affect admin connections through Voyager or FTP. **Default:** yes
`com2-login <yes | no>`	Specifies login on the serial port ttyd1 com2 that may be connected to an external modem. **Default:** no
`com3-login <yes | no>`	Specifies login on the serial port ttyd2 com3 that may be connected to an external modem. **Default:** no
`com4-login <yes | no>`	Specifies login on the serial port ttyd3 com4 that may be connected to an external modem. **Default:** no
`echo <yes | no>`	Specifies echo service, which sends back any data received by the platform to the originating source. **Default:** no

`discard <yes | no>`	Specifies discard service, which discards any data received by the platform. **Default:** no
`chargen <yes | no>`	Specifies chargen service, which sends back any data without regard to input. The data sent is a repeating sequence of printable characters. **Default:** no
`daytime <yes | no>`	Specifies daytime service, which sends the current date and time as a character string without regard to the input. **Default:** no
`time <yes | no>`	Specifies time service, which sends back the time, in seconds, since midnight January 1, 1900 the originating source. The value is sent as a binary number. **Default:** no

`modem com4 <country code>`	For the Ositech Five of Clubs card, use the following country codes. For the US, enter 22; for Canada, enter 20; for Australia, enter 1; for Belgium, enter 2; for Denmark, enter 3; for Finland, enter 4; for France, enter 5; for Germany, enter 6; for Greece, enter 16; for Iceland, enter 99; for Ireland, enter 7; for Italy, enter 8; for Luxembourg, enter 9; for the Netherlands, enter 10; for Portugal, enter 12; for Spain, enter 13; for Sweden enter 14; for Switzerland, enter 25; and for United Kingdom, enter 16. For the Ositech Five of Clubs II card, use of the following country codes. For the US, enter B5; for Canada, enter 20, for Australia, enter 09; for Belgium, enter 0F; for Denmark, enter 31; for Finland, enter 3C; for France, enter 3D; for Germany, enter 42; for Greece, enter 46; for Iceland, enter 57; for Italy, enter 59; for Luxembourg, enter 69; for the Netherlands, enter 7B; for Norway, enter 82; for Portugal, enter B8; for Spain, enter A0; for Swedent, enter A5; for Switzerland, enter A6, and for United Kingdom, enter B4.

Licenses

To purchase a license or increase your limit on your current license, call your Nokia representative or call (650) 625-2000 or (888) 477-4566 or email sales@iprg.nokia.com.

Configuring Software Licenses

Use the following commands to add a software license to your platform.

```
set licensing
        bgp-key license_key
        dvmrp-rip-key license_key
        dvmrp-ospf-key license_key
        igrp-key license_key
        dvmrp-key license_key
```

Use the following commands to delete a software license from your platform.

```
delete licensing
        bgp-key
        dvmrp-rip-key
        dvmrp-ospf-key
        igrp-key
        dvmrp-key
```

Use the following commands to show your software licenses.

```
show
        licensing
        licensing bgp-key
        licensing dvmrp-rip-key
        licensing dvmrp-ospf-key
        licensing igrp-key
        licensing dvmrp-key
```

Arguments

bgp-key *license_key*	Enables BGP feature.
dvmrp-rip-key *license_key*	Enables DVMRP with RIP feature.
dvmrp-ospf-key *license_key*	Enables DVMRP with OSPF feature.

`igrp-key` *`license_key`*	Enables IGRP feature.
`dvrmp-key` *`license_key`*	Enables DVMRP feature.

IPsec Commands (IPSO Implementation)

This section describes the CLI commands you use to configure the IPSO implementation of IP Security (IPsec) on your system and to view current settings. IPsec is the industry standard that ensures the construction of secure virtual private networks (VPNs).

Use the commands in this section to configure and view different IPsec entities including filters, proposals, rules, policies, keys, X509 certificates, tunnels, and transports. You can also configure IPsec options such as debugging level and hardware acceleration.

Note
Because the IP2250 and IP2255 platforms require the use of Check Point's SecureXL, these platforms do not support IPSO's implementation of IPsec.

General IPsec Commands

Use the following command to turn off IPsec. This command clears the complete IPsec tree in the system on which the command is issued. Use it only when you want to turn off IPsec on the system.

```
set ipsec clear
```

Use the following command to show a summary of all IPsec configuration, including configured rules, proposals, filters, policies, and other IPsec information:

```
show ipsec all
```

Proposal Commands

IPsec proposals specify the encryption and authentication algorithms, ordered by priority, available to the gateway at the remote endpoint of the IPsec tunnel. This list is shared among all IPsec tunnel interfaces. The two types of proposals are:

- Encapsulating security payload (ESP)
- Authentication header (AH).

> **Note**
> AH proposals do not use an encryption algorithm.

Each proposal indicates one transform along with its mode. You cannot configure transform parameters, such as cipher-key length. The IPSO CLI implementation does not support user configuration of IKE proposals, which are derived from the IPsec proposals.

Use the following `add` command to create a proposal. If you specify only the proposal name, the proposal is ESP with SHA1 authentication and DES encryption. Use the `set` command to modify one or more properties of an existing proposal. You cannot modify the proposal type.

> **Note**
> You cannot use the word *all* for the value of a command variable represented by *`name`*. In other words, do not use *all* to identify a proposal, policy, or rule.

```
add ipsec proposal name
```

```
add ipsec proposal name type
        esp auth <sha1 | md5> crypto <des | 3des | blowfish | null>
        ah auth <sha1 | md5>

set ipsec proposal name
        auth <sha1 | md5>
        crypto <des | 3des | blowfish | null>
```

Arguments

`proposal` *name*	Specifies a unique identifier for the proposal: • Format: single word, no spaces allowed • Example: 3des-md5
`type <esp | ah>`	Specifies the type of proposal to add. **Default:** esp
`auth <sha1 | md5>`	Specifies the algorithm used to generate and verify a hash for every packet. **Default:** md5
`crypto <des | 3des | blowfish | null>`	Specifies the algorithm used to encrypt the packets. AH proposals cannot have an encryption algorithm. **Default:** des

Use the following command to list the attributes of one or all proposals:

```
show ipsec proposal
        all
        name
```

Arguments

`proposal all`	Shows information about all proposals.

`proposal` *`name`*	Shows information about a specified proposal.

Use the following command to delete the specified proposal. If the proposal is linked to a policy, the proposal cannot be deleted unless the policy is deleted first (see "Policy Commands" on page 303).

```
delete ipsec proposal
        all
        name
```

Arguments

`proposal all`	Deletes all proposals. You cannot delete proposals that are linked to a policy.
`proposal` *`name`*	Specifies the name of the proposal to delete.

Filter Commands

Filters specify the list of addresses or subnet addresses that the IPsec engine matches against the address field of an IP packet. If the address matches, IPsec is applied. This list is shared among all IPsec tunnel interfaces. Every connection can have two or more filters since each filter indicates either source or destination options.

Use the following `add` command to create a filter on an IPv4 or IPv6 network. If you do not specify a protocol or port, the default value for these fields is *any* and no port. Use the `set` command to modify one or more properties of an existing filter.

```
add ipsec filter name
        address ip_address mask <0—32>
               proto <tcp | udp | icmp | any>
               port <0—65535>
        address6 ip6_address mask <0—128>
               proto <tcp | udp | icmp | any>
               port <0—65535>

set ipsec filter name
        address ip_address mask <0—128>
        address6 ip6_address mask6 <0—128>
        proto <tcp | udp | icmp | any>
        port <0—65535>
```

Arguments

`filter name`	Specifies a name for the filter: • Format: single word, no spaces allowed • Example: 3des-md5
`address ip_address`	Specifies the IP address of the desired subnetwork with host bits as zeroes (0): • Example: 10.2.0.0
`mask <0—32>`	Specifies the mask length for the network.
`address6 ip6_address`	Specifies the IPv6 address of the desired subnetwork: • Example: 1000:50:32::
`mask6 <0—128>`	Specifies the mask length for the network.
`proto <tcp | udp | icmp | any>`	Specifies the protocol of the packet to filter. **Default:** any
`port <0—65535>`	Specifies the port number of the packet to filter, if appropriate, ignored otherwise.

Use the following command to show information about the specified filter:

```
show ipsec filter
        all
        name
```

Arguments

`filter all`	Shows information about all filters.
`filter` *`name`*	Shows information about a specified filter.

Use the following command to delete the specified filter. If the filter is linked to any connection, the filter cannot be deleted unless the connection is deleted first.

```
delete ipsec filter
        all
        name
```

Arguments

`filter all`	Delete all filters.
`filter` *`name`*	Specifies a name of the filter to delete.

Certificate Commands

Peer systems use certificates to authenticate each other. To do this, a system presents a trusted certificate to its peer to prove that it is what it claims. The IPSO IPsec implementation allows installation of x509 certificates, which can be used in IKE negotiation. The certificates can be either certification authority (CA) certificates or device certificates.

The CA certificates are trusted certificates. If the certificate for a device is found to be signed by the same CA as another device, the other device is trusted too. Device certificates are used as the identity of a host and are presented to peers during IKE negotiation.

Use the following `add` command to install an x509 CA or device certificate on a system. Use the `set` command to install a new certificate under the same x509 certificate name. None of the attributes for the new certificate is derived from the old certificate. You must specify all parameters when you install the new certificate. The `realm`, `username`, and `password` parameters are optional and valid only when you specify `URL` as the certificate source.

```
add ipsec x509cert name type <dev | ca> source
        pem_file name
        url url url
               realm name [user username password password]

set ipsec x509cert name source
        pem file name
        url url url
               realm name [user username password password]
```

Arguments

`x509` *name*	Specifies the name of the certificate.
`type <dev | ca>`	Specifies the type of certificate to create. You can use all uppercase or all lowercase: • dev: device certificate • ca: CA certificate
`source <pem | url>`	Specifies how to obtain certificate. You can use all uppercase or all lowercase: • pem: obtain as input by using a simple file with PEM encoding. • url: obtain directly from a URL to be installed in the local host.

`file` *name*	Specifies the file, including the path, containing the CA certificate or the signed DEV certificate. **Note:** This parameter is valid only if the source is PEM
`url` *url*	Specifies the location where certificate can be found. The certificate can be downloaded from an HTTP, FTP, LDAP or file server: • Format: Standard URL format. • Example: http://test.acme.com/dev1.cert • Example: ftp://test.acme.com/dev1.cert • Example: file:///tmp/dev1.cert • Example: ldap://test.acme.com/ cn=dev1.acme.com?pem_x509?sub This argument does not have a default value.
`realm` *name*	If the URL you specified is under access control, you must specify the name of the realm containing the certificate. This field is optional
`user` *username*	Specifies the username for access to the realm. This field is optional.
`password` *password*	Specifies the password associated with the username. This field is optional.

Use the following command to show information about the specified certificate or all certificates:

```
show ipsec x509cert
        all
               options <attribs | content | decoded>
        name
               options <attribs | content | decoded>
```

Arguments

`x509cert all`	Shows information about all certificates.
`x509` *name*	Shows information about the specified certificate.
`options <attribs | content | decoded>`	Specifies the information to show: • attribs: shows location and other attributes of certificate. • content: shows the original content of the certificate. • decoded: shows the decoded human-readable form the certificate. **Default:** attribs

Use the following command to delete the specified certificate. Using the keyword `all` instead of a certificate name deletes all certificates.

```
delete ipsec x509cert
        all
        name
```

Arguments

`x509cert all`	Deletes all certificates.
`x509` *name*	Specifies the name of the certificate to delete.

IPSO can generate a set of RSA/DSA public and private keys. The public key can be included in a certificate request, along with other attributes, enabling the certificate to be signed by a CA.

Use the following `add` command to generate a certificate request to be sent to a CA. Use the following `set` command to generate a new certificate request under the same name. None of the attributes for the new certificate are derived from the old certificate. You must specify all parameters when you set the certificate request.

```
add ipsec x509certreq name key-len <512 | 768 | 1024> sig-algo
   <dsa | rsa> pass-phrase phrase country country state state locality
   locality org name org-unit name dns-name name
        ip-address ip_address email email_address
        ip-address6 ip6_address email email_address

set ipsec x509certreq name key-len <512 | 768 | 1024> sig-algo
   <dsa | rsa> pass-phrase phrase country country state state locality
   locality org name org-unit name dns-name name
        ip-address ip_address email email_address
        ip-address6 ip6_address email email_address
```

Arguments

`x509certreq` *name*	Specifies the name of the certificate request.
`key-len <512 | 768 | 1024>`	Specifies how large (and therefore how secure) your newly generated private key is. It is specified in bits. **Default:** 1024
`sig-algo <dsa | rsa>`	Specifies the algorithm to use to generate the keys. **Default:** rsa
`pass-phrase` *phrase*	Specifies the passphrase to use to protect private key.
`country` *country*	Specifies the two-letter code indicating your country, for example, US.
`state` *state*	Specifies the name of your state or province, for example, California.

`locality` *`locality`*	Specifies the locality (town or city) name, for example, San Francisco.
`org` *`name`*	Specifies the name of your company or organization, for example, Nokia.
`org-unit` *`name`*	Specifies the name of your subunit within your company or organization, for example, Nokia Engineering Department.
`dns-name` *`name`*	Specifies the common DNS name (fully qualified domain name), for example: www.dns.nokia.com.
`ip-address` *`ip_address`*	Specifies a valid IPv4 address: • Example: 192.168.50.5
`ip-address6` *`ip6_address`*	Specifies a valid IPva6 address: • Example: 1000:50:32::2
`email`	Specifies the email address to contact the person responsible for this system or for its certificate, for example, webmaster@engineering.nokia.com. The CA sends notification to this address.

Use the following command to show information about the specified certificate request or all certificate requests:

```
show ipsec x509certreq
        all
               options <attribs | content | decoded>
        name
               options <attribs | content | decoded>
```

Arguments

`x509certreq all`	Shows information about all x509 certificate requests.
`x509certreq` *`name`*	Shows information about the specified certificate request.
`options <attribs | content | decoded>`	Specifies the information to show: • attribs: shows location and other attributes of certificate. • content: shows the original content of the certificate. • decoded: shows the decoded human-readable form of the certificate. **Default:** attribs

Use the following command to delete all x509 certificate requests or a specific certificate request:

```
delete ipsec x509certreq
        all
        name
```

Arguments

`x509certreq all`	Deletes all x509 certificate requests.
`x509certreq` *`name`*	Specifies the name of the certificate request to delete.

Policy Commands

Policy defines the use of filters together with a list of IPsec proposals, ordered by priority. These policies can be applied to one or several IPsec tunnel interfaces, which creates a secure tunnel.

Use the following `add` command to create a policy, which can be used to create a connection. Use the `set` command to modify one or more properties of an existing policy. You can set the properties together or individually. Lifetime values must be set to the same value between peers when negotiation is initiated. If they are not set the same, IPSO IPsec may deny the negotiation.

```
add ipsec policy name proposal name priority integer
        psk secret_key
               life-sec <0—700000>
               life-mb <0—65000>
               ike-group <1 | 2 | 5>
               pfs-group <1 | 2 | 5 | none>
        x509cert name
               life-sec <0—700000>
               life-mb <0—65000>
               ike-group <1 | 2 | 5>
               pfs-group <1 | 2 | 5 | none>

set ipsec policy name
        proposal name priority integer
        psk secret_key
        x509cert name
        life-sec <0—700000>
        life-mb <0—65000>
        ike-group <1 | 2 | 5>
        pfs-group <1 | 2 | 5 | none>
```

Arguments

`policy` *name*	Specifies the name of the policy to add.

`proposal` *`name`*	Specifies the proposal with which to associate this policy. The proposal must already exist.
`priority` *`integer`*	Specifies the priority of the policy. A higher value indicates lower priority.
`psk` *`secret_key`*	Specifies the preshared-secret key. It must be between 8 and 256 characters.
`x509cert` *`name`*	Specifies the name of the predefined device certificate.
`life-sec <0–700000>`	Specifies the number of seconds from security association (SA) creation to start the rekeying. This parameter is optional. **Default:** 300
`life-mb <0–65000>`	Specifies the amount of megabytes of data transferred before rekeying is started. This parameter is optional. **Default:** 100
`ike-group <1 | 2 | 5>`	Specifies the Diffie-Hellman group used in establishing phase-1 ISAKMP SA. This parameter is optional. **Default:** 2
`pfs-group <1 | 2 | 5 | none>`	Specifies the Diffie-Hellman group used in establishing phase-2 IPsec SA. You can specify `none` to disable PFS in phase-2. This parameter is optional. **Default:** 2

Use the following command to view information about the specified policy or all policies:

```
show ipsec policy
        all
        name
```

Arguments

`policy all`	Shows information about all policies.
`policy name`	Shows information about a specified policy.

Use the following command to delete the specified policy or all policies. If you specify a proposal, this removes the proposal from the named policy. If a policy is linked to any connection, it cannot be deleted unless the connection is deleted first.

```
delete ipsec policy
        all
        name
               proposal name
```

Arguments

`policy all`	Delete all policies
`policy name`	Specifies the name of the policy to delete.The policy must already exist. If you do not specify a proposal, the policies matching the *name* are deleted.
`proposal name`	Specifies the proposal to delete from the policy. The proposal must already exist.

Rule Commands

IPsec rules specify the set of actions to be performed on packets matching the selectors. A rule can be specified in tunnel mode or in transport mode.

For a tunnel mode rule, if you specify a separate logical interface, it is automatically created. Additionally, the remote endpoint is also added to the classifier so that the reverse classifier lookup can be done in the input path. For both tunnel and transport mode rules, the destination filters are added to the IPsec classifier so that packets are directed to the engine in the output path.

Use the following `add` command to add an IPsec tunnel mode rule for an IPv4 network or an IPv6 network. You can specify any combination (or all) of the third-level parameters in the same `add` command. Use the `set` command to modify one or more properties of an existing rule. Once you add a rule, you cannot change the rule mode or indicate whether a separate logical interface is needed for a tunnel mode rule.

```
add ipsec rule name mode tunnel
        local-address ip_address remote-address ip_address
               policy name
               src-filter name
               dst-filter name
               inc-end-points <on | off>
               logical-interface <on | off>
               hello-prot <on | off> [hello-inv <0—21666> dead-inv <0—
                  65000>]
        local-address6 ip6_address remote-address6 ip6_address
               policy name
               src-filter name
               dst-filter name
               inc-end-points <on | off>
               logical-interface <on | off>
               hello-prot <on | off> [hello-inv <0—21666> dead-inv <0—
                  65000>]
```

```
set ipsec rule name
        local-address ip_address
        remote-address ip_address
        local-address6 ip6_address
        remote-address6 ip6_address
        policy name
        src-filter name
        dst-filter name
        inc-end-points <on | off>
        hello-prot <on | off>
        hello-inv <0—21666>
        dead-inv <0—65000>
```

Use the following `add` command to add an IPsec transport mode rule. Use the `set` command to change one or more values for the existing transport mode rule.

```
add ipsec rule name mode transport
        policy name
        src-filter name
        dst-filter name

set ipsec rule name
        policy name
        src-filter name
        dst-filter name
```

Arguments

`rule` *name*	Specifies the name of the rule to add.
`mode <tunnel | transport>`	Specifies the rule mode. The rest of the command depends on this value. Trasnport mode rules accept values only for policy name, source filter, and destination filter. **Default:** tunnel

`local-address` *`ip_address`*	Specifies the local IP address used as the tunnel endpoint. It must be an address of another interface configured for this system: • Example: 10.2.0.1
`remote-address` *`ip_address`*	Specifies the IP address of the multicast router at the remote end of the tunnel. It cannot be the local address of any interface of this system: • Example: 10.2.0.2
`local-address6` *`ip6_address`*	Specifies the local IPv6 address used as the tunnel endpoint. It must be an address of another interface configured for this system: • Example: 1000:50:32::1
`remote-address6` *`ip6_address`*	Specifies the IPv6 address of the multicast router at the remote end of the tunnel. It cannot be the local address of any interface of this system: • Example: 1000:50:32::2
`policy` *`name`*	Specifies an existing policy for the rule.
`src-filter` *`name`*	Specifies an existing source filter for the rule.
`dst-filter` *`name`*	Specifies an existing destination filter for the rule.
`inc-end-points <on | off>`	Specifies whether the tunnel end points are included in the filter. **Default:** off
`logical-interface <on | off>`	Specifies whether a separate logical interface is needed for this tunnel mode rule. **Default:** on

`hello-prot <on | off>`	Specifies whether the Hello protocol is running on the tunnel. Determines the end-to-end connectivity of a tunnel interface, modifying the link state if necessary. **Default:** off
`hello-inv <0—21666>`	Specifies the time in seconds between Hello packets. The Hello Protocol must be running to set this value. **Default:** 1
`dead-inv <0—65000>`	Specifies the time in seconds before the interface is considered down. The Dead interval should be 3 times or more than the Hello interval. The Hello protocol must be running to set this value. **Default:** 3

Use the following command to display information about a specified rule. Also, you can show all tunnel mode rules, all transport mode rules, or all rules.

```
show ipsec rule
        all [mode <tunnel | transport>]
        name
```

Arguments

`rule all`	Shows information about all rules.
`mode <tunnel | transport>`	Specifies whether to show all tunnel mode rules or all transport mode rules. This parameter is optional. **Default:** tunnel
`rule` *`name`*	Shows information about a specific rule.

Use the following command to detach a policy from a rule, delete a source or destination filter from the specified rule, or delete the specified rule entirely. Also, you can delete all rules or all rules of a specified mode.

```
delete ipsec rule
        all [mode <tunnel | transport>]
        name
               policy name
               src-filter name
               dst-filter name
```

Arguments

`mode <tunnel | transport>`	Specifies the rule mode. This allows you to delete all tunnel mode rules or all transport mode rules. **Default:** tunnel
`rule` *name*	Specifies the name of the rule.
`policy` *name*	Specifies the name of the policy to be detached from the rule.
`src-filter` *name*	Specifies an existing source filter for the rule.
`dst-filter` *name*	Specifies an existing destination filter for the rule.
`rule all`	Delete all rules. If you do not use the optional mode parameter, you delete all tunnel and transport mode rules.

Miscellaneous IPsec Commands

Use the following command to configure various IPsec functionality:

```
set ipsec
        log-level <error | debug | info>
        hardware-accl <on | off>
        allow-interfaceless-tunnels <on | off>
```

Arguments

`log-level <error | debug | info>`	Specifies the IPsec logging level. This affects the logging IPsec sends to SYSLOG. **Default:** error
`hardware-accl <on | off>`	Turns hardware acceleration on or off for all interfaces. When you turn hardware acceleration on, it is enabled for all the interfaces. **Default:** off
`allow-interfaceless-tunnels <on | off>`	Specifies whether to allow IPsec tunnels that are not associated with logical interfaces. In a new installation this flag is turned off. **Default:** off

Use the following commands to add or delete a specified LDAP server to the IPsec lookup list. The certificate revocation lists (CRLs) for the IPsec CA certificates are obtained from the specified LDAP server.

```
add ipsec ldap url

delete ldap url
```

Arguments

`ldap` *`url`*	Specifies the URL of the LDAP server.

Use the following commands to view miscellaneous information about IPsec properties:

```
show ipsec
        log-level
        ldap
        hardware-accl
        allow-interfaceless-tunnels
```

AAA

Use the following group of commands to configure AAA.

Viewing AAA Configuration

Use the following command to view the AAA configuration.

```
show aaa all
```

Configuring Service Modules

Use the following command to create a new AAA service and associate it with a service profile.

```
add aaa service name profile name
```

Use the following command to delete a service entry.

```
delete aaa service name
```

Use the following command to change the configuration of an existing AAA service and associate it with a new service profile.

```
set aaa service name profile name
```

Use the following commands to view the service module configuration and a service profile entry of a particular service.

```
show aaa
        services
        service name
```

Arguments

`service` *name*	The name of an application or service that is to use AAA.
`profile` *name*	A service profile entry created by the command `add aaa profile` and may have associated authentication, account, and session profiles. The following profiles are included in the IPSO operating system: • base_prof_httpd • base_prof_login • base_prof_other • base_prof_snmpd • base_prof_sshd

Configuring Service Profiles

Use the following commands to create new service profiles and associate authentication, account, and session profiles to new or existing services profiles or add to existing service profiles.

```
add aaa profile name
        authprofile name acctprofile name sessprofile name
        authprofile name
        acctprofile name
        sessprofile name
```

Use the following command to delete profiles or authentication, account and session profiles grouped under a service profile.

```
delete aaa
        profile name
        profile name authprofile name
        profile name acctprofile name
        profile name sessprofile name
        profile name auth-priority name
        profile name acct-priority name
        profile name sess-priority name
```

Use the following commands to set the order in which multiple authentication, account or session profiles will run for a given service profile.

```
set aaa profile name
        authprofile name auth-priority integer
        acctprofile name acct-priority integer
        sessprofile name sess-priority integer
```

Use the following commands to view particular service profile entries.

```
show aaa
        profiles
        profile name
        profile name authcount
        profile name acctcount
        profile name sesscount
        profile name authprofiles
        profile name auth-priority integer
        profile name acctprofiles
        profile name acct-priority integer
        profile name sessprofiles
        profile name sess-priority integer
```

Arguments

`profile` *`name`*	A service profile entry created by the command `add aaa profile` *`name`* and may have associated authentication, account, and session profiles. The following profiles are included in the IPSO operating system: • base_prof_httpd • base_prof_login • base_prof_other • base_prof_snmpd • base_prof_sshd
`authprofile` *`name`*	An authentication profile entry. The following authentication profiles are included in the IPSO operating system: • base_httpd_authprofile • base_login_authprofile • base_other_authprofile • base_snmpd_authprofile • base_sshd_authprofile
`acctprofile` *`name`*	An account profile entry. The following account profiles are included in the IPSO operating system: • base_httpd_acctprofile • base_login_acctprofile • base_other_acctprofile • base_snmpd_acctprofile • base_sshd_acctprofile

`sessprofile` *`name`*	A session profile entry. The following session profiles are included in the IPSO operating system: • base_httpd_sessprofile • base_login_sessprofile • base_other_sessprofile • base_snmpd_sessprofile • base_sshd_sessprofile
`auth-priority` *`integer`* `acct-priority` *`integer`* `sess-priority` *`integer`*	A number indicating the priority of a given algorithm. When multiple algorithms are configured in an authentication, account or session profile for a given service profile, the priority number determines the order that the associated algorithms are tried. A smaller number indicates a higher priority.

Configuring Authentication Profiles

Use the following command to create an authentication profile entry.

```
add aaa authprofile name [authtype name authcontrol name]
```

Use the following command to delete an authentication profile entry.

```
delete aaa authprofile name
```

Use the following command to change authentication profile configurations.

```
set aaa authprofile name
        authtype name
        authcontrol name
```

Use the following command to view authentication profile configurations.

```
show aaa
        authprofiles
        authprofile name
        authprofile name authtype
        authprofile name authcontrol
```

Arguments

`authprofile` *`name`*	The name of a new authentication profile (add) or existing authentication profile (delete, set, show). The IPSO operating system includes the following authentication profiles: • base_httpd_authprofile • base_login_authprofile • base_other_authprofile • base_snmpd_authprofile • base_sshd_authprofile
`authtype` *`name`*	The name of the authentication algorithm. The IPSO operating system includes the following authentication algorithms: • HTTPD • PERMIT • RADIUS • ROOTOK • SECURETTY • SKEY • SNMPD • TACPLUS • UNIX

`authcontrol` *`name`*	The name of a control value. The control value determines how the results of multiple authentication algorithms are applied and when additional algorithms are invoked. A control value other than *required* is only effective when multiple authentication profiles are defined for a given service profile.The IPSO operating system includes the following control values: • required—The result is retained and the next algorithm is invoked. • requisite—A result of *error* is reported immediately and no further algorithms are invoked. Otherwise, the result is retained and the next algorithm is invoked. • sufficient—A result of *error* is ignored; If a previous algorithm has reported *error*, or the result of this algorithm is *error*, the next algorithm is invoked. • optional —A result of *error* is ignored and a result of success is retained. The next algorithm is always invoked. • nokia-server-auth-sufficient—Same as “sufficient,” except a result of *error* for this module is reported immediately and no further modules are invoked.

Configuring Account Profiles

Use the following command to create an account profile entry.

```
add aaa acctprofile name accttype name acctcontrol name
```

Use the following command to delete an account profile entry.

```
delete aaa acctprofile name
```

Use the following command to change account profile configurations.

```
set aaa acctprofile name
        authtype name
        authcontrol name
```

Use the following command to view account profile configurations.

```
show aaa
        acctprofiles
        acctprofile name
        acctprofile name accttype
        acctprofile name acctcontrol
```

Arguments

`acctprofile` *name*	The name of a new accounting profile (add) or existing accounting profile (delete, set, show). The IPSO operating system includes the following account profiles: • base_httpd_acctprofile • base_login_acctprofile • base_other_acctprofile • base_snmpd_acctprofile • base_sshd_acctprofile
`accttype` *name*	The name of the account management algorithm. The IPSO operating system includes the following account management algorithms: • PERMIT—(pam.permit.so.1.0) This algorithm returns PAM_SUCCESS when invoked. • UNIX—(pam.unix.acct.so.1.0) This algorithm provides the basic UNIX accounting mechanism by checking if the password is still valid. If the password is expired, the algorithm logs in appropriate messages. The algorithm also prompts for a password change if the password is going to expire soon.

`acctcontrol` *`name`*	The name of a control value. The control value determines how the results of multiple accounting algorithms are applied and when additional algorithms are invoked. A control value other than *required* is only effective when multiple accounting profiles are defined for a given service profile.The IPSO operating system includes the following control values: • required—The result is retained and the next algorithm is invoked. • requisite—A result of *error* is reported immediately and no further algorithms are invoked. Otherwise, the result is retained and the next algorithm is invoked. • sufficient—A result of *error* is ignored; If a previous algorithm has reported *error*, or the result of this algorithm is *error*, the next algorithm is invoked. • optional —A result of *error* is ignored and a result of success is retained. The next algorithm is always invoked.

Configuring Session Profiles

Use the following command to create a session profile entry.

```
add aaa sessprofile name sesstype name sesscontrol name
```

Use the following command to delete an session profile entry.

```
delete aaa sessprofile name
```

Use the following command to change session profile configurations.

```
set aaa sessprofile name
        sesstype name
        sesscontrol name
```

Use the following command to view session profile configurations.

```
show aaa
        sessprofiles
        sessprofile name
        sessprofile name sesstype
        sessprofile name sesscontrol
```

Arguments

`sessprofile` *name*	The name of a new session profile (add) or existing session profile (delete, set, show). The IPSO operating system includes the following session profiles: • base_httpd_sessprofile • base_login_sessprofile • base_other_sessprofile • base_snmpd_sessprofile • base_sshd_sessprofile
`sesstype` *name*	The name of the session management algorithm. The IPSO operating system includes the following session management algorithms: • PERMIT—(pam.permit.so.1.0) The algorithm returns PAM_SUCCESS when invoked. • UNIX—(pam.unix.sess.so.1.0) The algorithm logs a message to indicate that a session has started or stopped.

`sesscontrol` *name*	The name of a control value. The control value determines how the results of multiple session algorithms are applied and when additional algorithms are invoked. A control value other than *required* is only effective when multiple session profiles are defined for a given service profile.The IPSO operating system includes the following control values: • required—The result is retained and the next algorithm is invoked. • requisite—A result of *error* is reported immediately and no further algorithms are invoked. Otherwise, the result is retained and the next algorithm is invoked. • sufficient—A result of *error* is ignored; If a previous algorithm has reported *error*, or the result of this algorithm is *error*, the next algorithm is invoked. • optional —A result of *error* is ignored and a result of success is retained. The next algorithm is always invoked.

Configuring RADIUS

Use the following command to configure RADIUS for use in a single authentication profile.

```
add aaa radius-servers authprofile name priority integer host IPv4
   address port integer <secret name | prompt-secret> timeout integer maxtries
   integer
```

Use the following command to delete a RADIUS configuration.

```
delete aaa radius-servers authprofile name priority integer
```

Use the following commands to change the configuration of a RADIUS entry.

```
set aaa radius-servers authprofile name priority integer
        host IPv4 address
        port integer
        secret name
        prompt-secret
        timeout integer
        maxtries integer
        new-priority integer
```

Use the following command to view a list of all servers associated with an authentication profile.

```
show aaa radius-servers authprofile name list
```

Use the following commands to view a RADIUS configuration.

```
show aaa radius-servers authprofile name priority integer
        host
        port
        timeout
        maxtries
        new-priority
```

Arguments

`authprofile` *name*	An authentication profile entry. The following authentication profiles are included in the IPSO operating system: • base_httpd_authprofile • base_login_authprofile • base_other_authprofile • base_snmpd_authprofile • base_sshd_authprofile

`priority` *integer*	A number indicating the priority of the server. When you configure multiple servers, the priority determines which should be tried first. A smaller number indicates a higher priority. No default. You must enter a value.
`host` *IPv4 address*	The IP address of the RADIUS server in dot-delimited format.
`port` *1-65535*	The UDP port to contact on the server host. You determine the correct value by the cofiguration of your RADIUS server. Common values are 1812 (specified by the standard) and 1645 (nonstandard but used traditionally). *port* also takes the keyword *unset*, which indicates that no value is set in the database so the default is used. **Default:** 1812
`secret` *name*	The "shared secret" used to authenticate the RADIUS server and the local client to each other. The same value should be configured on your RADIUS server.
`prompt secret`	Prompts you to enter the "shared secret" during the run of the command.
`timeout` *integer*	The number of seconds to wait, after contacting the server, for a response. The default value 3 seconds.
`maxtries` *integer*	The number of attempts to make to contact the server. **Default:** 3 (includes the first attempt, so a value of 3 means two retries)

`new-priority` *integer*	Changes the priority number of the server.
`list`	Displays a list of all servers associated with an authentication profile.

Configuring TACPLUS

Use the following command to configure TACPLUS for use in a single authentication profile.

```
add aaa tacplus-servers authprofile name priority integer host IPv4
   address port integer <secret name | prompt-secret> timeout integer maxtries
   integer
```

Use the following command to delete a TACPLUS configuration.

```
delete aaa tacplus-servers authprofile name
   priority integer
```

Use the following commands to change the configuration of a TACPLUS entry.

```
set aaa tacplus-servers authprofile name priority integer
        host IPv4 address
        port integer
        secret name
        prompt-secret
        timeout integer
        new-priority integer
```

Use the following command to view a list of all servers associated with an authentication profile.

```
show aaa radius-servers authprofile name list
```

Use the following commands to view a TACPLUS configuration.

```
show aaa tacplus-servers authprofile name priority integer
        host
        port
        timeout
```

Arguments

`authprofile` *name*	An authentication profile entry. The following authentication profiles are included in the IPSO operating system: • base_httpd_authprofile • base_login_authprofile • base_other_authprofile • base_snmpd_authprofile • base_sshd_authprofile
`priority` *integer*	A number indicating the priority of the server. When you configure multiple servers, the priority determines which should be tried first. A smaller number indicates a higher priority. No default. You must enter a value.
`host` *IPv4 address*	The IP address of the TACPLUS server in dot-delimited format.
`port <1-65535>`	The UDP port to contact on the server host. You determine the correct value by the cofiguration of your TACPLUS server. Common values are 1812 (specified by the standard) and 1645 (nonstandard but used traditionally). *port* also takes the keyword *unset*, which indicates that no value is set in the database so the default is used. **Default:** 1812

`secret` *`name`*	The “shared secret” used to authenticate the TACPLUS server and the local client to each other. The same value should be configured on your TACPLUS server.
`prompt secret`	Prompts you to enter the “shared secret” during the run of the command.
`timeout` *`integer`*	The number of seconds to wait, after contacting the server, for a response. The default value 3 seconds.
`new-priority` *`integer`*	Changes the priority number of the server.
`list`	Displays a list of all servers associated with an authentication profile.

SSH

Use the following groups of commands to enable and configure the SSH service on your platform. By default the service is disabled.

Enabling/Disabling SSH Service

Use the following commands to enable, disable and show the status of SSH service.

```
set ssh server
        enable <0 | 1>

show ssh server
        enable
```

Arguments

`enable <0 | 1>`	The value of 0 disables SSH and the value of 1 enables SSH. **Default:** 1

Configuring Server Options

The commands in this section allow you to configure SSH server options.

Configuring Server Access Control

Use the following commands to configure who is allowed to log in to your platform.

```
set ssh server
        allow-groups name
        allow-users name
        deny-groups name
        deny-users name
        permit-root-login <yes | no | without-password>
```

Use the following commands to show login configurations.

```
show ssh server
        allow-groups
        allow-users
        deny-groups
        deny-users
        permit-root-login
```

Arguments

`allow-groups` *name*	Specifies that only a user whose primary or supplementary group name matches this pattern may log in. You may use wildcard characters ' * , ? and specify multiple user names or patterns separated by spaces. Numerical group indentifications are not recognized. Login is allowed for all groups by default.
`allow-users` *name*	Specifies that only a user whose login name matches this pattern may log in. You may use wildcard characters ' * , ? and specify multiple user names or patterns separated by spaces. Numerical user indentifications are not recognized. If the pattern takes the form USER@HOST then USER and HOST are checked separately. Login is allowed for all users by default.
`deny-groups` *name*	Specifies groups forbidden to log in. Each user has only one primary group, identified by the group id in the user's configuration.
`deny-users` *name*	Specifies users forbidden to log in.
`permit-root-login <yes | no | without-password>`	Specifies whether admin can log in using SSH. The *without-password* setting allows admin to log in, but not by using the password mode of authentication.

Configuring Server Authentication of Users

Use the following commands to configure the type of authentication the server will use to authenticate users.

```
set ssh server
        pubkey-authentication <0 | 1>
        password-authentication <0 | 1>
        rhosts-authentication <0 | 1>
        rhosts-rsa-authentication <0 | 1>
        rsa-authentication <0 | 1>
```

Use the following commands to show user authentication configurations.

```
show ssh server
        pubkey-authentication
        password-authentication
        rhosts-authentication
        rhosts-rsa-authentication
        rsa-authentication
```

Arguments

`pubkey-authentication <0 | 1>`	Specifies whether pure DSA authentication is allowed (for version 2 of the protocol only). **Default:** 1
`password-authentication <0 | 1>`	Allows authentication using the standard log in password. If this is set to 0, other forms of authentication may be used. This may require additional configuration. **Default:** 1
`rhosts-authentication <0 | 1>`	Specifies whether authentication using *rhosts* or *hosts.equiv* is sufficient. **Default:** 0
`rhosts-rsa-authentication <0 | 1>`	Specifies whether *rhosts* or *hosts.equiv* authentication is accepted when combined with successful RSA host authentication. **Default:** 0

`rsa-authentication <0 | 1>`	Specifies whether pure RSA authentication is allowed (for version 1 of the protocol only). **Default:** 1

Configuring User Login Environment

Use the following commands to configure the user environment.

```
set ssh server
        print-motd <0 | 1>
        use-login <0 | 1>
```

Use the following commands to show user environment configurations.

```
show ssh server
        print-motd
        use-login
```

Arguments

`print-motd <0 | 1>`	Specifies whether text from /etc/motd is displayed when the user logs in interactively. **Default:** 1
`use-login <0 | 1>`	Specifies whether the *login* utility is used for interactive logins. **Default:** 0

Configuring Server Protocol Details

Use the following commands to configure SSH protocols.

```
set ssh server
        ciphers name
        keepalives <0 | 1>
        listen-addr ip_address
        listen-addr2 ip_address
        port <1-65535>
        protocol <1 | 2 | 1,2>
        server-key-bits <512 | 640 | 768 | 864 | 1024>
```

Use the following commands to show SSH protocol configurations.

```
show ssh server
        ciphers
        keepalives
        listen-addr
        listen-addr2
        port
        protocol
        server-key-bits
```

Arguments

`ciphers` *name*	The following list of ciphers are used as the default setting: • 3des-cbc • blowfish-cbc • arcfour • cast128-cbc • aes128-cbc • aes192-cbc • aes256-cbc When you use the set command to specify a cipher, or series of ciphers separated by commas, all other ciphers not specified are turned off.

`keepalives <0 | 1>`	Specifies whether the system should send keepalive messages to the other side. **Default:** 1
`listen-addr` *`ip_address`*	Specifies the IP address on which to listen for incoming connections. Use dot-delimited format.
`listen-addr2` *`ip_address`*	Specifies the IP address on which to listen for incoming connections. Use dot-delimited format.
`port <1-65535>`	Specifies the TCP port number on which the SSH server listens. The default port is 22.
`protocol <1 | 2 | 1,2>`	Specifies which of the two versions of the SSH protocols to support. Both versions 1 and 2 are on by default. When you use the set command to specify a protocol version, the other protocol version is turned off. **Default:** 1,2
`server-key-bits <512 | 640 | 768 | 864 | 1024>`	Defines the number of bits in the server key. **Default:** 768

Configuring Service Details

Use the following commands to configure service details.

```
set ssh server
        gateway-ports <0 | 1>
        ignore-rhosts <0 | 1>
        ignore-user-known-hosts <0 | 1>
        key-regeneration-time integer
        login-grace-time integer
        max-starups integer
```

Use the following commands to show service detail configurations.

```
show ssh server
        gateway-ports
        ignore-rhosts
        ignore-user-known-hosts
        key-regeneration-time
        login-grace-time
        max-starups
```

Arguments

`gateway-ports <0 | 1>`	Specifies whether remote hosts are allowed to connect to ports forwarded for the client. The value for no is zero (0). **Default:** 0
`ignore-rhosts <0 | 1>`	Specifies that *.rhosts* and *.shosts* files will not be used in authentication. This is relevant only if *rhosts* authentication or *rhosts+RSA* authentication is enabled. The value for yes is 1. **Default:** 1
`ignore-user-known-hosts <0 | 1>`	Specifies whether the *known_hosts* file is used during *rhosts+RSA* authentication. The value for no is zero (0). **Default:** 0
`key-regeneration-time` *integer*	Specifies the amount of time in seconds when the server key is automatically regenerated. Default is 3600 seconds. A value of 0 means the key is never regenerated.
`login-grace-time` *integer*	Specifies the amount of time in seconds for the user to log in. Default is 600 seconds. A value of 0 means unlimited time.

`max-startups` *integer*	Specifies the maximum number of unauthenticated connections which will be allowed at one time. Default is 10.

Configuring Server Implementation

Use the following commands to set the level of verbosity of sshd logged messages and to specify whether to check file modes and ownership of files and directories at log in.

```
set ssh server
        log-level name
        strict-modes <0 | 1>
```

Use the following commands to show service detail configurations.

```
show ssh server
        log-level
        strict-modes
```

Arguments

`log-level` *name*	Specifies the verbosity level used when logging messages from *sshd*. The following are the values you can designate from least to most verbose: • QUIET • FATAL • ERROR • INFO • VERBOSE • DEBUG Default is INFO.

`strict-modes <0 | 1>`	Specifies whether to check file modes and ownership of certain files and directories before allowing the user to log in. The value for yes is 1. **Default:** 1

Configuring and Managing SSH Key Pairs

The commands in the following section allow you to configure SSH key pairs and to obtain the current public keys.

Creating and Viewing Host Keys

Use the following commands to generate new host keys.

```
set ssh hostkey
        v1 size integer
        v2 <dsa | rsa> size integer
```

Use the following commands to view the current public host keys.

```
show ssh hostkey
        v1
        v2 <dsa | rsa> [ssh2-format]
```

Arguments

`v1`	Generates a new RSA v1 public/private key pair that will be used by the SSH server to authenticate itself to clients using SSH protocol version 1.

`v2 <dsa | rsa>`	Generates a new RSA v2 or DSA public/ private key pair that will be used by the SSH server to authenticate itself to clients using SSH protocol version 2.
`size` *integer*	Specifies the size of the key in bits. 1024 bits is recommended. Other common key sizes are 512, 640, 768, and 864. The key must be at least 512 bits.
`ssh2-format`	Displays the RSA v2 or DSA public key in SSHv2 format. If you omit this argument, the public key is displayed in OpenSSH format.

Generating New User Identity Keys

Use the following commands to generate new user identity keys that will be used to authenticate users with other SSH servers.

```
set ssh identity
        v1 user name size integer <passphrase string
           | prompt-passphrase>
        v2 <dsa | rsa> user name size integer <passphrase string
           | prompt-passphrase>
```

Use the following commands to obtain the public identity keys.

```
show ssh identity
        v1 user name
        v2 <dsa | rsa> [ssh2-format]
```

Arguments

`v1`	Creates a new RSA v1 public/private key pair that will be used to authenticate the given user to other SSH servers when connecting using SSH protocol version 1
`v2 <dsa | rsa>`	Creates a new DSA or RSA v2 public/private key pair that will be used to authenticate the given user to other SSH servers when connecting using SSH protocol version 2.
`user` *`name`*	Identifies the user for which you want to create a new key pair.
`size` *`integer`*	Specifies the size of the key in bits. 1024 bits is recommended. Other common key sizes are 512, 640, 768, and 864. The key must be at least 512 bits.
`passphrase` *`string`*	Provides an optional passphrase that is used to encrypt the private key. The user must enter this passphrase when using the key. If you use this argument, the passphrase text is echoed on the screen as you enter it.
`prompt-passphrase`	Tells the system to prompt you for the optional passphrase after you enter the command. The passphrase is not displayed on the screen as you enter it.

Managing Authorized Keys

Use the following commands to add authorized public keys for users, enabling the users to log in to their accounts using their keys.

```
add ssh authkeys
        v1 user name bits integer exponent integer modulus string
           [comment name]
        v2 <dsa | rsa> user name <openssh-format string | ssh2-format
           file name> [comment name]
```

Use the following commands to delete authorized keys.

```
delete ssh authkeys
        v1 user name id number
        v2 <dsa | rsa> user name id number
```

Use the following commands to view authorized keys configured for user accounts.

```
show ssh authkeys
        v1 user name < list | id number>
        v2 <rsa | dsa> user name < list | id number>
```

Arguments

`v1`	Specifies that the authorized key being added is an RSA v1 public key.
`v2 <dsa | rsa>`	Specifies that the authorized key being added is either a DSA or RSA v2 public key.
`user` *`name`*	Identifies the user account for which you are authorizing the key.
`bits` *`integer`* `exponent` *`integer`* `modulus` *`string`*	Provides the RSA v1 key as follows: • `bits` *`integer`*–key size, for example 1024. • `exponent` *`integer`*–usually the second number making up an RSA v1 key, for example 35. • `modulus` *`string`*–a very long number that is usually the third value of an RSA key.

`<openssh-format` *`string`* | `ssh2-format file` *`name`*`>`	Provides the DSA or RSA v2 key in either OpenSSH or SSHv2 format. • `openssh-format` *`string`*—the text of key in OpenSSH format, which is a long string of characters, numbers, and punctuation. Omit the “ssh-dss” or “ssh-rss” label that might come before the key. • `ssh2-format file` *`name`*—the file on the local system that contains the SSHv2 formatted key.
`comment` *`string`*	Specifies an optional comment which can be used to help identify the key.
`id` *`number`*	The identification number associated with a particular key. The ID value is assigned by the system and is used to specify the key for viewing or deletion.
`list`	Shows ID values and any optional comments associated with authorized keys.

Voyager Web Access (SSL)

Use the following groups of commands to configure Voyager web access service.

Enabling SSL Voyager Web Access

Use the following commands to enable SSL web access and encryption.

```
set voyager
        daemon-enable <0 | 1>
        port <1-65535>
        ssl-port <1-65535>
        ssl-level <0-168>
```

Use the following commands to view the SSL configuration.

```
show voyager
        port
        ssl-port
        ssl-level
        daemon-enable
```

Arguments

`daemon-enable <0 | 1>`	Enables and disables web configuration for the platform. **Default:** 1
`port <1-65535>`	Specifies the port number on which the Voyager web configuration tool can be accessed when *not* using SSL-secured connections. If you change the port number, you will have to change the URL used when accessing Voyager from http://hostname/ to http://hostname:PORTNUMBER/ **Default:** 80
`ssl-port <1-65535>`	Specifies the port number on which the Voyager web configuration tool can be accessed when using SSL-secured connections. If you change the port number, you will have to change the URL used when accessing Voyager from https://hostname/ to https://hostname:PORTNUMBER/ **Default:** 443

`ssl-level <0-168>`	Specifies the required level of security for Voyager web connections. The value zero (0) indicates that SSL-secured connections will not be used. Setting the level of encryption requires remote connections to use a level of encryption *at least* as strong as the one you specify. The following are the standard encryption levels: • 40-bit • 56-bit • 128-bit • 168-bit (Triple-DES) Once you specify a level of encryption, you must change your URL from http://hostname/ to https://hostname/ to access your platform. **Default:** 0

Generating a Certificate and Private Key

Use the following command to generate a certificate and its associated private key. To better ensure your security, you should generate the certificate and private key over a trusted connection.

```
generate voyager ssl-certificate key-bits <512 | 768| 1024> <passphrase
   name | prompt-passphrase> country name state-or-province name
   locality name organization name organizational-unit name common-name
   name email-address name <cert-file path | cert-request-file path>
   key-file path
```

Arguments

`key-bits <512 | 768 | 1024>`	Specifies how large your newly generated private key will be in bits. Larger sizes are generally considered more secure. **Default:** 1024

`passphrase` *name*	Specifies a string that this tool will use to encrypt your new private key. Using this syntax will echo your passphrase as you type. If you do not wish to use a passphrase, enter an empty one as ("").
`prompt-passphrase`	Specifies a string that this tool will use to encrypt your new private key. Using this syntax will not echo your passphrase as you type.
`country` *name*	Specifies a two letter code indicating your country, for example, US. This is a required entry.
`state-or-province` *name*	Specifies the *name* of your state or province. This is a required entry.
`locality` *name*	Specifies the name of your city or town, for example Sunnyvale. If you do not wish to use a passphrase, enter an empty one as ("").
`organization` *name*	Specifies the name of your company or organization, for example Worldwide Widgets. This is a required entry.
`organizational-unit` *name*	Specifies the name of a subunit within your company or organization. If you do not wish to use a passphrase, enter an empty one as ("").
`common-name` *name*	Identifies where the certificate will go. The name is most commonly the fully qualified domain name for your platform, for example, www.ship.wwwidgets.com. If you are generating a request for a certificate authority, the issuer may impose a different standard.
`email-address` *name*	Specifies an e-mail address that could be used for contacting the person responsible for platform and its certificate, for example, “webmaster@ship.wwwidgets.dom”

`cert-file path`	Specifies a file that will receive a certificate. The keyword should be followed by the path name to a file on the IPSO system. Use absolute pathnames.The certificate will be signed with a SHA-1 hash.
`cert-request-file path`	Specifies a file that will receive a certificate request. The keyword should be followed by the path name to a file on the IPSO system. Use absolute pathnames. The request will be signed with a SHA-1 hash.
`key-file path`	Specifies a file that will receive, a private key. The keyword should be followed by the path name to a file on the IPSO system. Use absolute pathnames.

Installing a Certificate and Private Key

Use the following commands to copy a certificate and its associated private key in the /var/etc/voyager_ssl_server.crt and /var/etc/voyager_ssl_server.key files. Copying the certificate and private key to these files makes them available to establish SSL-secure web connections.

```
set voyager ssl-certificate
        cert-file path key-file path <passphrase
           name | prompt-passphrase>
```

Arguments

`cert-file` *`path`*	Specifies a file that contains the certificate you want to copy. The keyword should be followed by the path name to the file on the IPSO system. Use absolute pathnames.

`key-file` *`path`*	Specifies a file that contains the private key you want to copy. The keyword should be followed by the path name to a file on the IPSO system. Use absolute pathnames.
`passphrase` *`name`*	Enter the passphrase you used when generating the certificate and private key or certificate request. Using this syntax will echo your passphrase as you type.
`prompt-passphrase`	Prompts you to enter the passphrase you used whtn generating the certificate and private key or certificate request. Using this syntax will not echo your passphrase as you type.

Password and Account Management

Use these commands to set policies for managing user passwords and accounts. The features included in password and account management are a global and comprehensive way to manage password policy. For more information about these features, see the "Managing Security and Access" chapter of the *Nokia Network Voyager Reference Guide*.

To manage individual user accounts, see "Managing System Users" on page 350 and "Managing Roles" on page 353 for more information.

To view password and account management configuration, use these commands.

```
show password-controls
        min-password-length
        complexity
        palindrome-check
        history-checking
        history-length
        password-expiration
        expiration-warning-days
        expiration-lockout-days
        force-change-when
        deny-on-fail enable
        deny-on-fail failures-allowed
        deny-on-fail allow-after
        deny-on-nonuse enable
        deny-on-nonuse allowed-days
        all
```

To change password and account management configuration, use these commands.

```
set password-controls
        min-password-length <6-128>
        complexity <2-4>
        palindrome-check <on | off>
        history-checking <on | off>
        history-length <1-1000>
        password-expiration <never | 1-1827>
        expiration-warning-days <1-366>
        expiration-lockout-days <never | 1-1827>
        force-change-when <no | password | first-password>
        deny-on-fail enable <on | off>
        deny-on-fail failures-allowed <2-1000>
        deny-on-fail allow-after <60-604800>
        deny-on-nonuse enable <on | off>
        deny-on-nonuse allowed-days <30-1827>
```

Arguments

`min-password length <6-128>`	Specifies the minimum number of characters of a password that is to be allowed for users or SNMP users. Does not apply to passwords that have already been set. **Default:** 6
`complexity <1-4>`	Specifies how complex users must make their passwords. Password characters are divided into four types: upper case alphabetic (A-Z), lower case alphabetic (a-z), digits (0-9), and other (everything else such as !, #,.&.). The following is a description of the values: • 1 disables complexity checking. • 2 requires that passwords are comprised of a minimum of two character types, for example, abcABC. • 3 requires that passwords are comprised of a minimum of three character types, for example, ab1ABC. • 4 requires that passwords are comprised of a minimum of four character types, for example, ab1AB#. **Default:** 3
`palindrome-check <on | off>`	Checks for passwords that are read the same left to right or right to left, such as racecar or the phrase straw warts. **Default:** on
`history-checking <on | off>`	Enables a check for passwords being reused. Password history checking does not apply to SNMP users. **Default:** on

`history-length <1-1000>`	Specifies the number of past passwords that will be kept and checked against for reuse for each user. **Default:** 10
`password-expiration <never | 1-1827>`	Specifies the number of days since the last password change before a user is required to set a new password. Once the specified number of days have passed without a password change, the password expires and the user is required to change password again the next time the user logs in. The value “never” disables the feature. Password expiration applies to regular users, not to SNMP users. **Default:** never
`expiration-warning-days <1-366>`	Specifies the number of days before a password expires that a user starts receiving warnings that the password is due to expire. **Default:** 7 days
`expiration-lockout-days <never | 1-1827>`	Specifies the number of days after password expiration that a user is locked out if the user has not set a new password. **Default:** never

`force-change-when <no | password | first-password>`	Specifies whether to force users to change their passwords. • `no` disables forcing a user to change an assigned password at login. • `password` forces a user to change a password after an administrator sets it with the `set user` command or with the Network Voyager User Management page. The forced password change does not apply to passwords set by the user through a self-password change. • `first-password` forces a new user to change password from the initial password assigned by the administrator when the user account is created. **Default:** no
`deny-on-fail enable <on | off>`	Locks out a user after a configurable number of failed logins. When you enable this feature, keep in mind that it leaves you open to a denial of service attack by someone issuing unsuccessful login attempts. **Default:** off
`deny-on-fail failures-allowed <2-1000>`	Sets the number of failed logins a user can have before being locked out. **Default:** 10
`deny-on-fail allow-after <60-604800>`	Sets the number of seconds a user has to wait before being able to log in after being locked out because of failed logins. **Default:** 1200
`deny-on-nonuse enable <on | off>`	Enables locking out users who have not logged in during a configurable amount of time. **Default:** off

`deny-on-nonuse allowed-days <30-1827>`	Sets the number of days after a user has last logged in before they are locked out. **Default:** 365

Users and Roles Management

Use the commands in "Managing System Users" to add and delete new system users. The system has default admin and monitor users. You can use the commands in "Changing the Admin and Monitor Password" to change the password for admin and monitor.

To create an IPSO cluster, you must create a cluster configuration on each system that participates in the cluster. When you do this, the system creates a cadmin user on each of these systems, and you must create the same password for the cadmin user on each system. (You use the cadmin user to manage a cluster.) You can use the commands described in this section to manage the password for the cadmin user.

This command set also includes commands for configuring an S/Key, a one-time password system that authenticates remote telnet and FTP users.

Managing System Users

Use these commands to add new users and to set and change user passwords, user ID, group ID, home directory, and default shell.

For information on setting SNMP users, see "Managing SNMP Users".

Use the following commands to view configuration and conditions.

```
show users
```

```
show user username
        force-password-change
        lock-out
```

Using the following command to delete an existing user.

```
delete user username
```

Use the following commands to add or modify user accounts.

```
set user username
        passwd
        newpass passwd
        info string
        uid <0-65535>
        gid <0-65535>
        homedir unix_path_name
        shell string
        homepage tcl_script_name
        force-password-change <on | off>
        lock-out off

add user username
        uid <0-65535> homedir Unix path name
```

Note
You can use the add user command to add new users, but you must use the `set user name passwd` command to set the password and allow the user to log on to the system.

Arguments

`user username`	Specifies the new user name or an existing user name. The valid characters are alphanumeric characters, dash (-), and underscore (_). Range: 1-32 characters

`passwd`	Starts a password change dialog. You will be asked to enter a new password for the user and then asked to verify it by re-entering it. The password you enter will not be visible on terminal.
`newpass` *`passwd`*	Specifies a new password for the user. If you use this keyword to change the password, you will not be asked to verify the new password and the password you enter is visible on the terminal.
`info` *`string`*	Specifies a string describing the specified users. This string is used as the sender's name in any email sent by this user. Use alphanumeric characters and the space key only.
`uid <0-65535>`	Specifies the specified user's user ID, which is used to identify the user's permissions.
`gid <0-65535>`	Specifies the ID for the primary group to which a user belongs. Use the group management commands to specify membership in other groups.
`homedir` *`Unix_path_name`*	Specifies the user's home directory, where the user is placed on login. Enter the full Unix path name. If the directory doesn't already exist, it is created. The home directory for all users must be in /var/emhome/.
`shell` *`string`*	Specifies the user's shell, which is invoked when the user logs in. The default shell is /bin/csh. To change the shell, enter the new shell path name. Consult the /etc/shells file for valid login shells.
`homepage` *`tcl_script_name`*	This argument is not supported. Its use will not cause an error but it has no effect.
`force-password-change <on | off>`	On forces the user to change password the next time they log in. Off overrides any condition forcing the user to change password.

`lock-out off`	Clears any lockout conditions that are present on the user account and allows the user to log in.

For information on removing access mechanism permissions from a user, see the `delete rba user` commands below.

Managing Roles

To view existing role configurations, use these commands.

```
show rba
        all
        role rolename
        roles
        user username
        users
```

To add a role use the following command.

```
add rba role rolename domain-type <System | Cluster | MRI> [readonly-
   features featurelist readwrite-features featurelist]
```

To grant or change the access-mechanisms assigned to a user, use the following command.

```
add rba user username access-mechanisms <Voyager | CLI>
```

To assign a role to a user on a system which does not have clustering or MRI enabled, use the following command.

```
add rba user username roles rolename
```

To assign a role to a user on a system which has clustering or MRI enabled, use the following command.

```
add rba user username role rolename domains <System | Cluster | MRI>
```

Arguments

`role rolename`	Specifies the name of the role.
`domain-type <System | Cluster | MRI>`	Specify the role type. System roles apply only to this machine, cluster roles apply to each node in the cluster, and MRI roles apply to specified MRIs.
`[readonly-features featurelist readwrite-features featurelist]`	List each feature which you want to include in the role as having either read-only or read/write access. Separate each with a comma and do not use spaces. For a list of available features, use the tab completion in the CLI.
`access-mechanisms <Voyager | CLI>`	Assign users privilege to use Network Voyager or the CLI.
`domains <System | Cluster | MRI>`	Specifies the domain type of the role.

To remove features from a role, use this command.

```
delete rba role rolename [readonly-features featurelist
   readwrite-features featurelist}
```

To remove privileges for access mechanisms from a user, use this command.

```
delete rba user username access-mechanisms <Voyager | CLI>
```

To remove a role assignment from a user, use this command.

```
delete rba user username roles rolename
```

To remove access to an MRI from a user, use this command.

```
delete rba user username role rolename domains MRI MRIid
```

Arguments

`rba user` *`username`*	Specifies the user.
`roles` *`rolename`*	Specifies the name of the role. You can enter a comma separated list of roles if you want to remove more than one from the user.
`domain-type <System | Cluster | MRI>`	Specify the role type. System roles apply only to this machine, cluster roles apply to each node in the cluster, and MRI roles apply to specified MRIs.
`access-mechanisms <Voyager | CLI>`	Users privilege to use Network Voyager or the CLI.
`MRI` *`MRIid`*	Specifies the MRI to be removed from the user's access. You can include more than one MRI id number in a comma separated list.

Changing the Admin and Monitor Password

Use the following commands to change the password for admin and monitor. Admin and Monitor are default users. Typically, you set the initial passwords for admin and monitor at system startup.

> **Note**
> Only users with admin user permissions can change the user and admin passwords.

`set user <admin | monitor> newpass` *`passwd`*

`set user <admin | monitor> passwd`

Arguments

`<admin | monitor>`	Specifies whether to change the admin or the monitor password.
`newpass` *`passwd`*	Specifies the new password. If you use this keyword, you will not be asked to confirm the new password and the password you enter is visible on the terminal.
`passwd`	Specifies that you want to change the password. If you use this keyword, you will be prompted for the new password and then asked to verify the password by re-entering it.The password you enter is not visible on the terminal.

Configuring S/Key for Admin and Monitor

Use the following commands to configure S/Key for admin and monitor. S/Key is a one-time password system that authenticates telnet and FTP users.

```
set skey
        user <admin | monitor> mode <disabled | allowed | required>
        user <admin | monitor> key
        user <admin | monitor> currpass passwd secret string
        pass-phrase seed value sequence value
        pass-phrase seed value sequence value secret string
        pass-phrase seed value sequence value count value
        pass-phrase seed value sequence value count value secret
           string
```

Arguments

`user <admin | monitor> mode <disabled | allowed | required>`	Specifies whether to disable, allow, or require S/Key authentication for admin or monitor users.

`user <admin | monitor> key currpass` *`passwd`* `secret` *`string`*	Specifies an S/Key password for either admin or monitor users. The command requires you to enter the current admin or montior password for authentication purposes. Enter a string that contains alphanumeric values only as the secret.
`pass-phrase seed` *`value`* `sequence` *`value`*	Generates one S/Key pass-phrase using the seed and sequence values provided. The command prompts you to enter a secret
`pass-phrase seed` *`value`* `sequence` *`value`* `count` *`value`*	Generates count S/Key pass-phrases using the seed and sequence values provided. The command prompts you to enter a secret.
`pass-phrase seed` *`value`* `sequence` *`value`* `secret` *`string`*	Generates one S/Key pass-phrase using the seed, sequence, and secret values provided.
`pass-phrase seed` *`value`* `sequence` *`value`* `count` *`value`* `secret` *`string`*	Generates count S/Key pass-phrases using the seed, sequence, and secret values provided.

Show Commands

Use the following commands to view current users, their settings and permissions.

```
show users

show user name

show skey
        all
        user <admin | monitor>
        user <admin | monitor> sequence
        user <admin | monitor> seed
        user <admin | monitor> mode
```

Arguments

`users`	Displays all users, their user IDs, primary group IDs, user shells, home directories, and home pages.
`user` *`name`*	Displays information about the specified user, including the user ID, primary group ID, user shell, home directory and home page.

Arguments

`skey all`	Displays whether S/Key is enabled or disabled for both admin and monitor and the configured sequence and seed values if S/Key is enabled.
`skey user <admin | monitor>`	Displays whether S/Key is enabled or disabled for the specified user and the configured sequence and seed values if S/Key is enabled.
`skey user <admin | monitor> sequence`	Displays the configured sequence value for the specified user.
`skey user <admin | monitor> seed`	Displays the configured seed value for the specifed user
`skey user <admin | monitor> mode`	Displays whether S/Key is enabled or disabled for the specified user.

Group Management

The commands in this section allow you to manage groups.

Managing Groups

Use the following commands to create and delete groups and to add and remove members.

```
set group string gid <100-65530>

add group string gid <100-65530>

delete group string

add group string member username

delete group string member username
```

Arguments

`group string gid <100-65530>`	Specifies to create a new group and the group's ID. Enter a string of alphanumeric characters of 1 to 8 characters long. The name must be unique on your sytstem, and the numeric ID you specify must be unique on your system. Note that ID ranges 0-99 and 65531-65535 are reserved for system use. If you specify a value within these ranges, an error message is displayed.
`group string member username`	Specifies to add an existing user to an existing group, including users admin and monitor.

Show Commands

Use the following commands to view existing groups and group members.

```
show groups

show group string
```

Arguments

`groups`	Displays names of all existing groups, including wheel—the admin and root group—and the corresponding group ID and members.
`group string`	Displays information about the specified group name, including the group ID and members.

VPN Acceleration

If you have a VPN accelerator card installed in your system, you can use CLI commands to enable or disable it and view information.

Configuring VPN Acceleration

Use the following commands to disable or enable a VPN accelerator card.

```
set cryptaccel <disable | enable>
```

Displaying VPN Accelerator Information

Use the following commands to display VPN accelerator status or statistics.

```
show cryptaccel <status | statistics>
```

VPN accelerator status information includes the following:

- Device ID
- Status (up or down)

VPN accelerator statistics information includes the following:

- Contexts (created and current)
- Packets (received, dropped, and processed)
- Bytes (received, dropped, and processed)
- Errors (received digest, random number, buffer alignment, device, memory, context, and packet header)

9 Routing Commands

This chapter describes the routing commands that you can enter from the CLI prompt.

Route Map Commands

Route maps are used to control which routes are accepted and announced by dynamic routing protocols. Use route maps to configure inbound route filters, outbound route filters and to redistribute routes from one protocol to another.

You can define route maps only using the CLI, this feature is not available in Network Voyager.

Route maps support both IPv4 and IPv6 protocols, including RIP, BGP, RIPng, OSPFv2, and OSPFv3. BGP4++ policy can only be specified using route maps. For the other protocols, you can use either route maps or the Route Redistribution and Inbound Route Filters features that you configure using Network Voyager. Route map for import policy corresponds to Inbound Route Filters; route map for export policy corresponds to Route Redistribution.

Note
Route maps offer more configuration options than the based configuration for route redistribution and inbound route filters. They are not functionally equivalent.

Protocols can use route maps for redistribution and Network Voyager settings for inbound route filtering and vice versa. However, if one or more route maps are assigned to a protocol (for import or export) any corresponding Network Voyager configuration (for route redistribution or inbound route filters) is ignored.

Each route maps includes a list of match criteria and set statements. You can apply route maps to inbound, outbound, or redistribution routes. Routes are compared to the match criteria, and all the actions specified in the set criteria are applied to those routes which meet all the match conditions. You can specify the match conditions in any order. If you do not specify any match conditions in a route map, that route map then matches all routes.

You define route maps, then assign them to protocols for export or import policy for that protocol. Route maps take precedence over voyager based configuration.

To create a route map, use CLI commands to specify a set of criteria that must be matched for the command to take effect. If the criteria are matched, then the system executes the actions you specify. A route map is identified by name and also has an identifying number, an allow or restrict clause, and a collection of match and set statements.

There can be more than one instance of a route map (same name, different ID). The lowest numbered instance of a route map is checked first. Route map processing stops when either all the match criteria of some instance of the route map are satisfied, or all the instances of the particular route map are exhausted. If the match criteria are satisfied, the actions in the set section are performed.

Routing protocols can use more than one route map when you specify distinct preference values for each. The appropriate route map with lowest preference value is checked first.

Set Routemap Commands

Use the following commands to set a route map.

```
set routemap rm_name id <1-65535>
        <off|on>
        allow
        inactive
        restrict
```

Arguments

`routemap rm_name`	Specify the name of the routemap.
`id <1-65535>`	Specify the ID of the routemap. You can enter the keyword `default` or the default value `10`.
`<off|on>`	Use on to create a routemap, use off to delete a routemap.
`allow`	Specifies to allow routes that match the routemap.
`inactive`	Use this argument to temporarily disable a routemap. To activate the routemap, use the allow or restrict arguments.
`restrict`	Specifies that routes that match the routemap are not allowed.

To specify actions for a routemap, use the following commands.

Note
Some statements affect only a particular protocol. For information on which statements affect a given protocol, see the "Supported Route Map Statements by Protocol" section, below.
Also, the same parameter cannot appear both as a match and action statement in a routemap. These include Community, Metric, and Nexthop.

```
set routemap rm_name id id_number action
        aspath-prepend-count <1-25>
        community <append | replace | delete> [on|off]
        community <1-65535> as <1-65535> [on|off]
        community no-export [on|off]
        community no-advertise [on|off]
        community no-export-subconfed [on|off]
        community none [on|off]
        localpref <1-65535>
        metric <add|subtract> <1-16>
        metric igp [<add | subtract>] <1-4294967295>
        metric value <1-4294967295>
        nexthop <ip ipv4_address | ipv6 ipv6_address>
        precedence <1-65535>
        preference <1-65535>
        route-type <type-1 | type-2>
        remove action_name
```

Arguments

`routemap rm_name`	Specify the name of the routemap.
`id id_number`	Specify the ID of the routemap. You can enter the keyword `default` or the default value `10`.
`aspath-prepend-count`	Specifies to affix AS numbers at the beginning of the AS path. It indicates the number of times the local AS number should be prepended to the ASPATH before sending out an update. BGP only.
`community <append | replace | delete> [on|off]`	Operate on a BGP community string. A community string can be formed using multiple community action statements. You can specify keywords append, replace, or delete for the kind of operation to be performed using the community string. The default operation is append. BGP only.
`community <1-65535> as <1-65535> [on|off]`	Specifies a BGP community value.

`community no-export [on|off]`	Routes received that carry a communities attribute containing this value must not be advertised outside a BGP confederation boundary (a stand-alone autonomous system that is not part of a confederation should be considered a confederation itself)
`community no-advertise [on|off]`	Routes received that carry a communities attribute containing this value must not be advertised to other BGP peers.
`community no-export-subconfed [on|off]`	All routes received carrying a communities attribute containing this value MUST NOT be advertised to external BGP peers (this includes peers in other members autonomous systems inside a BGP confederation).
`community none [on|off]`	In action statement, this statement makes sense only if used with replace. This deletes all communities associated with a route so that the route has no communities associated with it. Using it with append or delete would be a no-operation. The CLI returns an error if you turn "none" on and other community values already defined or if "none" is defined and you add some other community value.
`localpref <1-65535>`	Set the local preference for BGP route. BGP only.
`metric [<add|subtract>] <1-16>`	Add to or subtract from the metric value. RIP and RIPng only.
`metric igp [<add | subtract> <1-4294967295>]`	Set metric to IGP metric value or add to or subtract from the IGP metric value. RIP/RIPng only.
`metric value <1-4294967295>`	Set the metric value. For RIP the metric is *metric*, for OSPF the metric is *cost*, and for BGP the metric is *MED*.
`nexthop <ip` *`ipv4_address`* `| ipv6` *`ipv6_address`*`>`	Set IPv4 or IPv6 Nexthop Address. BGP only. **Note**: The ipv6 address should not be a link-local address.

`precedence <1-65535>`	Sets the rank of the route. Precedence works across protocols. Use this setting to bias routes of one protocol over the other. The lower value has priority.
`preference <1-65535>`	Applies only to BGP. This is equivalent to the bgp weight (in Cisco terms) of the route. However, unlike Cisco, the route with lower value will be preferred. This value is only relevent for the local router.
`route-type <type-1 | type-2>`	Type of OSPF external route. The metric type of AS External route is set to the specified value. Only applies to routes redistributed to OSPF.
`remove` *`action_name`*	Remove the specified action from the routemap. For community, it removes all community statements.

To specify the criteria that must be matched for the routemap to take effect, use the following commands.

Note
Some statements affect only a particular protocol. For information on which statements affect a given protocol, see the "Supported Route Map Statements by Protocol" section, below.
Also, the same parameter cannot appear both as a match and action statement in a routemap. These include Community, Metric, and Nexthop.

```
set routemap rm_name id <1-65535> match
        as <1-65535> [on | off]
        aspath-regex ["regular_expression" | empty] origin
           <any | igp | incomplete>
        community <1-65535> as <1-65535> [on|off]
        community exact [on|off]
        community no-export [on|off]
        community no-advertise [on|off]
        community no-export-subconfed [on|off]
        community none [on|off]
        ifaddress <IPv4_addr | IPv6_addr> [on | off]
        interface interface_name [on | off]
        metric value <1-4294967295>
        neighbor <IPv4_addr | IPv6_addr> [on | off]
        network <IPv4_network | IPv6_network> / masklength
           <all | exact | off | refines>
        network <IPv4_network | IPv6_network> / masklength between
           masklength and masklength
        nexthop IPv4_addr | IPv6_addr [on | off]
        protocol <ospf2 | ospf2ase | ospf3 | ospf3ase | bgp | rip |
           ripng | static | direct | aggregate>
        route-type <type-1 | type-2 | inter-area | intra-area>
           [on | off]
        remove match_condition_name
```

Arguments

`as <1-65535> [on | off]`	Match the specified autonomous system number with the AS number of a BGP peer. For BGP only.
`aspath-regex ["<regular-expression>" | empty] origin <any | igp | incomplete>`	Match the specified aspath regular expression. For BGP only. **Note**: Enter the regular expression in quotation marks. Use the empty keyword to match a null ASpath.
`community <1-65535> as <1-65535> [on|off]`	Specify the BGP community value.

`community exact [on|off]`	Specify that the communities present in the route must exactly match all the communities in the routemap. In absence of the exact clause, the route can have other community values associated with it in addition to the ones contained in the routemap. You can have multiple community statements in a route map to form a community string.
`community no-export [on|off]`	All routes received that carry a communities attribute containing this value must not be advertised outside a BGP confederation boundary (a stand-alone AS that is not part of a confederation should be considered a confederation itself).
`community no-advertise [on|off]`	All routes received carrying a communities attribute containing this value must not be advertised to other BGP peers.
`community no-export-subconfed [on|off]`	All routes received carrying a communities attribute containing this value must not be advertised to external BGP peers (this includes peers in other members autonomous systems inside a BGP confederation).
`community none [on|off]`	Matches an empty community string, i.e. a route which does not have any communities associated with it. The CLI returns an error if you turn "none" on and other community values already defined, or if "none" is defined and you add some other community value.
`ifaddress <`*`IPv4_addr`* `|` *`IPv6_addr`*`> [on | off]`	Match the specified interface address. This can be either an IPv4 or an IPv6 address. There can be multiple if address statements.
`interface` *`interface_name`* `[on | off]`	Match the route if the nexthop lies on the specified interface name. There can be multiple interface statements.
`metric value <1-4294967295>`	Match the specified metric value.

`neighbor <IPv4_addr | IPv6_addr> [on | off]`	Match the neighbors IP address. BGP, RIP, or RIPng. There can be multiple neighbor statements.
`network <IPv4_network | IPv6_network> / masklength`	Use with the following keywords: • `all`: Match all networkds belonging to this prefix and masklength. This is a combination of exact and refines. • `between masklength and masklength`: Specify a range of masklengths to be accepted for the specified prefix. • `exact`: Match prefix exactly. • `off`: Delete the network match statement. • `refines`: Match networks with more specific mask lengths only. Matches only subnets. There can be multiple network match statements in a route map.
`nexthop <IPv4_addr | IPv6_addr> [on | off]`	Match the specified nexthop address.
`protocol <ospf2 | ospf2ase | ospf3 | ospf3ase | bgp | rip | ripng | static | direct | aggregate>`	Match the specified protocol. Use this for route redistribution.

`route-type <type-1 | type-2 | inter-area | intra-area> [on|off]`	As a match statement in routemap for export policy, it can be used by any protocol to redistribute OSPF routes. If route-type of inter-area or intra-area is specified, the protocol match condition should be set to ospf2 or ospf3 and if route-type of type-1 or type-2 is specified, then protocol match condition should be set to ospf2ase or ospf3ase. While exporting OSPF ASE routes to other protocol, if metric match condition is set but route-type match condition is not set, it will try to match the metric value for both type-1 and type-2 routes. There can be multiple route-type match statements.
`remove` *`match_condition_name`*	Remove the specified match condition from the routemap. For match conditions which can have multiple match statements (such as network, neighbor), this argument removes all of them.

Show Routemap Commands

```
show routemap rm_name <all | id VALUE>

show routemaps
```

Routemap Protocol Commands

To assign routemaps to protocols, use the following commands. The preference value specifies which order the protocol will use each routemap.

```
set <ospf | rip | ipv6 ospfv3 | ipv6 ripng> export-routemap rm_name
   preference VALUE on

set <ospf | rip | ipv6 ospfv3 | ipv6 ripng> import-routemap rm_name
   preference VALUE on
```

To turn a routemap off, use the following commands.

```
set <ospf | rip | ipv6 ospfv3 | ipv6 ripng> export-routemap rm_name
   off

set <ospf | rip | ipv6 ospfv3 | ipv6 ripng> import-routemap rm_name
   off
```

To view routemaps assigned to protocols, use the following command.

```
show <ospf | rip | ipv6 ospfv3 | ipv6 ripng> routemap
```

To set BGP routemaps for export and import policies, use the following commands.

> **Note**
> BGP supports both IPv4 and IPv6 routes; use the family option to specify for which address family the routemap will be used. Default is inet. To use for IPv6 routes the family should be set to inet6 or inet-and-inet6.

```
set bgp external remote-as <1-65535> export-routemap rm_name
        off
        preference <1-65535> [family <inet | inet6 | inet-and-inet6>]
           on

set bgp external remote-as <1-65535> import-routemap rm_name
        off
        preference <1-65535> [family <inet | inet6 | inet-and-inet6>]
           on

set bgp internal export-routemap rm_name
        off
        preference <1-65535> [family <inet | inet6 | inet-and-inet6>]
           on

set bgp internal import-routemap rm_name
        off
        preference <1-65535> [family <inet | inet6 | inet-and-inet6>]
           on
```

```
show bgp routemap
```

Note
You cannot use routemaps in BGP confederations. To configure route filters and redistribution for BGP confederations, use the Inbound Route Filters and Route Redistribution pages in Network Voyager.

Supported Route Map Statements by Protocol

Some statements affect only a particular protocol, for example, matching the Autonomous System Number is applicable only to BGP. If such a condition is in a routemap used by OSPF, the match condition is ignored. Any non-applicable match conditions or actions are ignored and processing is done as if they do not exist. A log message is generated in /var/log/messages for any such statements.

Note
The same parameter cannot appear both as a match and action statement in a routemap. These include Community, Metric, and Nexthop.

RIP/RIPng

- Import Match conditions: Neighbor, Network, Interface, Ifaddress, Metric, Neighbor, Nexthop.
- Import Actions: Precedence, Metric Add/Subtract
- Export Match conditions when exporting from RIP/RIPng - Interface, Ifaddress, Metric, Network, Nexthop
- Export Match Conditions when redistributing using Protocol match: According to the protocol from which route is being redistributed.
- Export Actions when exporting from RIP/RIPng - Metric Add/Subtract
- Export Actions when redistributing - Metric Set

OSPFv2/OSPFv3

- Import Match conditions: Network (Route Prefix)
- Import Actions: Precedence
- Export Match conditions when other protocols redistribute OSPF routes : Network, Interface, Ifaddress, Metric, Route-type, Nexthop
- Export Match conditions when OSPF redistributes routes from other protocols: Conditions supported by that protocol
- Export Actions when redistributing to AS External: Metric, Route-type

BGP

- Import Match conditions: Network (Route Prefix), AS number, ASPATH Regular Expression/Origin, Community String, Neighbor (BGP Peer), Interface, Ifaddress, Metric (MED), Nexthop
- Import Actions: Precedence, LocalPref, Preference (Weight), Nexthop IP/IPv6
- Export Match conditions when exporting from BGP - Metric (MED), Network, Nexthop, Interface, Ifaddress, AS Number, ASPATH, Community String
- Export Match conditions when redistributing other route into BGP using Protocol Match statement - Conditions supported by that protocol.
- Export Actions - Community String (Append, Replace, Delete), Metric (MED) (Set, IGP, Add to IGP, Subtract from IGP), Nexthop IP/IPv6, Aspath Prepend Count

Redistributing Static, Interface, or Aggregate Routes

When redistributing **static** routes into BGP, OSPFv2/v3 or RIP/RIPng the following match conditions are supported:

- Network Prefix,
- Nexthop
- Interface
- Ifaddress
- Protocol (proto = static)

When redistributing **interface/direc**t routes into BGP, OSPFv2/v3 or RIP/RIPng the following match conditions are supported:

- Network Prefix
- Interface
- Ifaddress
- Protocol (proto = direct)

When redistributing **aggregate** routes into BGP, OSPFv2/v3 or RIP/RIPng the following match conditions are supported:

- Network Prefix
- Protocol (proto = aggregate)

Route Map Examples

Example 1

Redistribute interface route for eth3c0 into ospf, and set the ospf route-type to AS type-2 with cost 20.

```
set routemap direct-to-ospf id 10 on
set routemap direct-to-ospf id 10 match interface eth3c0
set routemap direct-to-ospf id 10 match protocol direct
set routemap direct-to-ospf id 10 action route-type type-2
set routemap direct-to-ospf id 10 action metric value 20

set ospf export-routemap direct-to-ospf preference 1 on
```

Example 2

Do not accept routes from RIP neighbor 10.1.2.3, accept routes from neighbor 10.1.2.4 as is, and for all other routes increment the metric by 2.

```
set routemap rip-in id 10 on
set routemap rip-in id 10 restrict
set routemap rip-in id 10 match neighbor 10.1.2.3

set routemap rip-in id 15 on
set routemap rip-in id 15 match neighbor 10.1.2.4

set routemap rip-in id 20 on
set routemap rip-in id 20 action metric add 2

set rip import-routemap rip-in preference 1 on
```

Example 3

Redistribute all static routes into BGP AS group 400. Set the MED value to 100, prepend our AS number to the aspath 4 times. If the route belongs to the prefix 10.0.0.0/8, do not redistribute. Send all BGP routes whose aspath matches the regular expression (100 200+) and set the MED value to 200.

```
set routemap static-to-bgp id 10 on
set routemap static-to-bgp id 10 restrict
set routemap static-to-bgp id 10 match protocol static
set routemap static-to-bgp id 10 match network 10.0.0.0/8 all

set routemap static-to-bgp id 15 on
set routemap static-to-bgp id 15 match protocol static
set routemap static-to-bgp id 15 action metric 100
set routemap static-to-bgp id 15 action aspath-prepend-count 4

set routemap bgp-out id 10 on
set routemap bgp-out id 10 match aspath-regex "(100 200+)"
   origin any
set routemap bgp-out id 10 action metric 200
```

Note
We do not need match protocol statement for routes belonging to the same protocol.

```
set bgp external remote-as 400 export-routemap bgp-out
   preference 1 family inet on
set bgp external remote-as 400 export-routemap static-to-bgp
   preference 2 family inet on
```

Example 4

Redistribute all OSPFv3 (internal and external) routes into BGP group 400, setting the outgoing community string to `[no-export, 200 as 100]`. For BGP IPv6 routes, send them with an empty community string. For all routes set the nexthop value to `3003::abcd:1012` (the address on the interface connecting to the peers).

Note
To exchange IPv6 routes in BGP the multiprotocol capability must be turned ON in BGP Configuration for the peer.

```
set routemap ospf3-to-bgp id 10 on
set routemap ospf3-to-bgp id 10 match protocol ospf3  (OSPF3
   INTERNAL ROUTES)
set routemap ospf3-to-bgp id 10 action community replace on
set routemap ospf3-to-bgp id 10 action community no-export on
set routemap ospf3-to-bgp id 10 action community 200 as 100 on
set routemap ospf3-to-bgp id 10 action nexthop ipv6
   3003::abcd:1012
```

```
set routemap ospf3-to-bgp id 20 on
set routemap ospf3-to-bgp id 20 match protocol ospf3ase  (FOR
   AS EXTERNAL ROUTES)
set routemap ospf3-to-bgp id 20 action community replace on
set routemap ospf3-to-bgp id 20 action community no-export on
set routemap ospf3-to-bgp id 20 action community 200 as 100 on
set routemap ospf3-to-bgp id 10 action nexthop ipv6
   3003::abcd:1012

set routemap bgp-out id 10 on
set routemap bgp-out id 10 action community replace on
set routemap bgp-out id 10 action community none on
set routemap ospf3-to-bgp id 10 action nexthop ipv6
   3003::abcd:1012

set bgp external remote-as export-routemap bgp-out preference
   1 family inet6 on
set bgp external remote-as export-routemap ospf3-to-bgp
   preference 2 family inet6 on
```

BGP

When you do initial configuration, set the router ID. You can also use the following commands to change the router ID.

```
set router-id
        default
        ip_address
```

Arguments

`default`	Selects the highest interface address when OSPF is enabled.

ip_address	Specifies a specific IP address to assign as the router ID. Do not use 0.0.0.0 as the router ID address. Nokia recommends setting the router ID rather than relying on the default setting. Setting the router ID prevents the ID from changing if the default interface used for the router ID goes down.

Use the following group of commands to set and view parameters for BGP.

```
set as
        as_number
        off
```

Arguments

as *as_number*	Specifies the local autonomous system number of the router. This number is mutually exclusive from the confederation and routing domain identifier. The router can be configured with either the autonomous system number or confederation number, not both. **Caution:** When you change the autonomous system number, all current peer sessions are reset and all BGP routes are deleted. include the multiple instance routing name if you have configured multiple routing instances.
as off	Disables the configured local autonomous system number.

External BGP

Use the following commands to configure external sessions of the protocol, that is, between routers in different autonomous systems.

```
set bgp external remote-as as_number
        <on | off>
        aspath-prepend-count <1-25 | default>
        description text
        local-address ip_address <on | off>
        virtual-address <on | off>
        outdelay <0-65535>
        outdelay off
```

Arguments

as_number <on | off>	Specifies the autonomous system number of the external peer group. Enter an integer from 1-65535.
aspath-prepend-count <1-25 | default>	Specifies the number of times this router adds to the autonomous system path on external BGP sessions. Use this option to bias the degree of preference some downstream routers have for the routes originated by this router. Some implementations prefer to select paths with shorter autonomous system paths. Default is 1.
description *text*	You can enter a brief text description of the group.
local-address *ip_address* <on | off>	Specifies the address used on the local end of the tcp connection with the peer group. The local address must be on an interface that is shared with the peer or with the peer's gateway when the gateway parameter is used.

`virtual-address <on | off>`	Specifies for this router to use the VRRP virtual IP address as the local endpoint for TCP connections. You must also configure a local address to enable this option. See the command above. You can configure this option only on a VRRP master. **Note**: You must use Monitored Circuit mode when configuring virtual IP support for BGP or any other dynamic routing protocol. Do not use VRRPv2 when configuring virtual IP support for BGP.
`outdelay <0-65535>`	Specifies the amount of time in seconds that a route must be present in the routing database before it is redistributed to BGP. The configured value applies to all peers configured in this group. This feature dampens route fluctuation. The value zero (0) disables this feature. **Default:** 0
`outdelay off`	Disables outdelay.

BGP Peers

Use the following commands to configure BGP peers. IPSO supports both IPv4 and IPv6 adddresses for BGP peers.

A BGP IPv6 address can be either link local or global scoped. If a link local address is used for peering, the outgoing interface must also be configured.

```
set bgp external remote-as as_number peer ip_address
        <on | off>
        med-out <0—4294967294 | default>
        accept-med <on | off>
        multihop <on | off>
        no-aggregator-id <on | off>
        holdtime <6—65535 | default>
        keepalive <2—21845 | default>
        ignore-first-ashop <on | off>
        send-keepalives <on | off>
        send-route-refresh [request | route-update] [ipv4 | ipv6
            | All] [unicast]
        accept-routes <all | none>
        passive-tcp <on | off>
        removeprivateas <on | off>
        authtype none
        authtype md5 secret secret
        throttle-count <0—65535 | off>
        ttl <1-255 | default>
        suppress-default-originate <on | off>
        log-state-transitions <on | off>
        log-warnings <on | off>
        trace bgp_traceoption <on | off>
        capability <default | ipv4-unicast | ipv6-unicast>
```

Arguments

`<on | off>`	Specifies a specific peer <ip_address> for the group.
`med-out <0-4294967294 | default>`	Specifies the multi-exit discriminator (MED) metric used as the primary metric on all routes sent to the specified peer address. This metric overrides the default metric on any metric specified by the redistribute policy. External peers uses MED values to decide which of the available entry points into an autonomous system is preferred. A lower MED value is preferred over a higher MED value. **Default:** 4294967294

`accept-med <on | off>`	Specifies that MED be accepted from the specified peer address. If you do not set this option, the MED is stripped from the advertisement before the update is added to the routing table.
`multihop <on | off>`	Enables multihop connections with external BGP peers more than one hop away. By default, external BGP peers are expected to be directly connected. This option can also be used for external load-balancing.
`no-aggregator-id <on | off>`	Specifies the router's aggregate attribute as zero (rather than the router ID value). This option prevents different routers in an AS from creating aggregate routes with different AS paths.
`holdtime <6-65535 | default>`	Specifies the BGP holdtime interval, in seconds, when negotiating a connection with the specified peer. If the BGP speaker does not receive a keepalive update or notification message from its peer within the period specified in the holdtime field of the BGP open message, the BGP connection is closed. **Default:** 180 seconds
`keepalive <2-21945 |default>`	The keepalive option is an alternative way to specify a holdtime value in seconds when negotiating a connection with the specified peer. You can use the keepalive interval instead of the holdtime interval. You can also use both intervals, but the holdtime value must be 3 times the keepalive interval value. **Default:** 60 seconds
`ignore-first-ashop <on | off>`	Specifies to ignore the first autonomous system number in the autonomous system path for routes learned from the corresponding peer. Set this option only if you are peering with a route server in transparent mode, that is, when the route server is configured to redistribute routes from multiple other autonomous systems without prepending its own autonomous system number.

`send-keepalives <on | off>`	Specifies for this router always to send keepalive messages even when an update message is sufficient. This option allows interoperability with routers that do not strictly adhere to protocol specifications regarding updates.
`send-route-refresh [request | route-update] [ipv4 | ipv6 | All] [unicast]`	Specifies that the router dynamically request BGP route updates from peers or respond to requests for BGP route updates.
`accept-routes <all | none>`	Specifies an inbound BGP policy route if one is not already configured. Enter *all* to specify accepting routes and installing them with an invalid preference. Depending on the local inbound route policy, these routes are then made active or inactive. Enter *none* to delete routes learned from a peer. This option saves memory overhead when many routes are rejected because no inbound policy exists.
`passive-tcp <on | off>`	Specifies for the router to wait for the specified peer to issue an open message. No tcp connections are initiated by the router.
`removeprivateas <on | off>`	Specifies that private AS numbers be removed from BGP update messages to external peers.
`authtype none`	Specifies not to use an authentication scheme between peers. Using an authentication scheme guarantees that routing information is accepted only from trusted peers. **Default:** none
`authtype md5 secret` *secret*	Specifies to use md5 authentication between peers. In general, peers must agree on the authentication configuration to and from peer adjacencies. Using an authentication scheme guarantees that routing information is accepted only from trusted peers.

`throttle-count <0-65535 | off>`	Specifies number of BGP updates to send at one time. This option limits the number of BGP updates when there are many BGP peers. Off disables the throttle count option.
`ttl <1-255 | default>`	Specifies the value of the TTL (time to live) parameter, the number of hops over which the external BGP multihop session is established. Configure this value only if the multihop option is enabled. **Default:** 64
`suppress-default-originate <on | off>`	Specifies NOT to generate a default route when the peer receives a valid update from its peer.
`log-state-transitions <on | off>`	Specifies for the router to log a message whenever a peer enters or leave the established state.
`log-warnings <on | off>`	Specifies for the rouer to log a message whenever a warning scenario is encountered in the codepath.

`trace` *`bgp_traceoption`* `<on | off>`	Specifies tracing options for your BGP implemenation. Log messages are saved in the var/log/isprd directory. Enter the following words to set each trace option. • *`packets`*—Trace all BGP packets to this peer. • *`open`*—Trace all BGP open messages to this peer. • *`update`*—Trace all BGP update messages to this peer. • *`keepalive`*—Trace all keepalive messages to this peer. • *`all`*—Trace all message types. • *`general`* —Trace message related to Route and Normal. • *`route`*—Trace routing table changes for routes installed by this peer. • *`normal`*—Trace normal protocol occurrences. Abnormal protocol occurrences are always traced. • *`state`*—Trace state machine transitions in the protocol. • *`policy`*—Trace application of the protocol and user-specified policy to routes being imported and exported.
`capability <default | ipv4-unicast | ipv6-unicast>`	Specifies capabilities setting. Default is IPv4 unicast.

BGP Confederations

Use the following commands to configure BGP confederations. You can configure a BGP confederation in conjunction with external BGP.

```
set bgp
        confederation identifier as_number
        confederation identifier off
        confederation aspath-loops-permitted <1-10>
        confederation aspath-loops-permitted default
        routing-domain identifier as_number
        routing-domain identifier off
        routing-domain aspath-loops-permitted <1-10>
        routing-domain aspath-loops-permitted default
        synchronization <on | off>
```

Arguments

`confederation identifier` *`as_number`*	Specifies the identifier for the entire confederation. This identifier is used as as the autonomous system number in external BGP sessions. Outside the confederation, the confederation id is the autonomous system number of a single, large autonomous system. Thus the confederation id must be a globally unique, typically assigned autonomous system number.
`confederation identifier off`	Disables the confideration identifier.
`confederation aspath-loops permitted <1-10>`	Specifies the number of times the local autonomous system can appear in an autonomous system path for BGP-learned routes. If this number is higher than the number of times the local autonomous sytem appears in an autonomous system path, the corresponding routes are discarded or rejected.
`confederation aspath loops-permitted default`	Specifies a value of 1.
`routing-domain identifier` *`as_number`*	Specifies the routing domain identifier (RDI) for this router. You must specify the RDI if you are using BGP confederations. The RDI does not need to be globally unique since it is used only within the domain of the confederation.

`routing-domain identifier off`	Disables the routing-domain identifier
`routing-domain aspath-loops-permitted <1-10>`	Specifies the number of times the local autonomous system can appear in an autonomous system path for BGP-learned routes. If this number is higher than the number of times the local autonomous sytem appears in an autonomous system path, the corresponding routes are discarded or rejected.
`routing-domain aspath-loops-permitted default`	Specifies a value of 1.
`synchronization <on | off>`	Enables IGP synchronization. Set this option On to cause internal and confederation BGP peers to check for a matching route from IGP protocol before installing a BGP learned route.

BGP Route Reflection

Use the following commands to configure BGP route reflection. You can configure route reflection as an alternative to BGP confederations. Route reflection supports both internal and external BGP routing groups.

```
set bgp
        cluster-id ip_address
        cluster-id off
        default-med <0-65535>
        default-med off
        default-route-gateway ip_address
        default-route-gateway off
```

Arguments

`cluster-id` *`ip_address`*	Specifies the cluster ID used for route reflection. The cluster ID default is that of the router id. Override the defulat if the cluster has more than one route reflector
`cluster-id off`	Disables the cluster ID.
`default-med <0-65535>`	Specifies the multi-exit discriminator (MED) metric used to advertise routes through BGP.
`default-med off`	Disables the specified MED metric.
`default-route-gateway` *`ip_address`*	Specifies the default route. This route has a higher rank than any configured default static route for this router. If you do not want a BGP peer considered for generating the default route, use the `peer <`*`ip_address`*`> suppress-default-originate on` command.
`default-route-gateway off`	Disables the configured default BGP route.

BGP Route Dampening

Use the following commands to configure BGP route dampening. BGP route dampening maintains a history of flapping routes and prevents advertising these routes. A route is considered to be flapping when it is repeatedly transitioning from available to unavailable or vice versa.

```
set bgp dampening
        <on | off>
        suppress-above <2-32>
        suppress-above default
        reuse-below <1-32>
        reuse-below default
        max-flat <3-64>
        max-flat default
        reachable-decay <1-900>
        reachable-decay default
        unreachable-decay <1-2700>
        unreachable-decay default
        keep-history <2-5400>
        keep-history default
```

Arguments

`<on | off>`	Specifies whether to enable or disable BGP route dampening.
`suppress-above <2-32>`	Specifies the value of the instability metric at which route suppression takes place. A route is not installed in the forwarding table or announced even if it reachable during the period that it is suppressed.
`suppress-above default`	Specifies an instability metric value for suppressing routes of 3.
`resue-below metric <1-32>`	Specifies the value of the instability metric at which a suppressed route becomes unsuppressed if it is reachable but currently suppressed. The value assigned to the reuse-below metric must be lower than the suppress-above value.
`reuse-below metric default`	Specifies an instability metric value for announcing previously suppressed routes of 2.

`nax-flap <3-64>`	Specifies the upper limit of the instability metric. The value must be greater than the suppress-above value plus 1. Each time a route becomes unreachable, 1 is added to the current instability metric.
`max-flat default`	Specifies the upper limit of the instability metric as 16.
`reachable-decay <1-900>`	Specifies the time for the instability metric to reach half of its value when the route is reachable. The smaller the value the sooner a suppressed route becomes reusable.
`reachable-decay default`	Specifies a value of 300.
`unreachable-decay <1-2700>`	Specifies the time for the instability metric to reach half its value when the route is NOT reachable. The value must be equal to or higher than the reachable-decay value.
`unreachable-decay default`	Specifies a value of 900
`keep-history <2-5400>`	Specifies the period for which route flapping history is maintained for a given route.
`keep-history default`	Specifies a value of 1800.

Internal BGP

Use the following commands to configure internal BGP sessions, that is, between routers within the same autonomous system.

```
set bgp internal
        <on | off>
        description text
        med <0-65535>
        med default
        outdelay <0-65535>
        outdelay off
        nexthop-self <on | off>
        local-address ip_address <on | off>
        virtual-address <on | off>
        interface [all | if_name] <on | off>
        protocol [all | bgp_internal_protocol] <on | off>
        peer ip_address peer_type <on | off>
        peer ip_address weight <0-65535>
        peer ip_address weight off
        peer ip_address no-aggregator id <on | off>
        peer ip_address holdtime <6-65535>
        peer ip_address holdtime default
        peer ip_address keepalive <2-21845>
        peer ip_address keepalive default
        peer ip_address ignore-first-ashop <on | off>
        peer ip_address send-keepalives <on | off>
        peer ip_address send-route-refresh [request | route-
          update] [ipv4 | ipv6 | All] [unicast]
        peer ip_address accept-routes all
        peer ip_address accept-routes none
        peer ip_address passive-tcp <on | off>
        peer ip_address authtype none
        peer ip_address authtype md5 secret secret
        peer ip_address throttle-count <0-65535>
        peer ip_address throttle count off
        peer ip_address log-state-transitions <on | off>
        peer ip_address log-warnings <on | off>
        peer ip_address trace bgp_traceoption <on | off>
        peer ip_address capability <default | ipv4-unicast | ipv6-
          unicast> <on | off>
```

Arguments

`<on | off>`	Specifies whether to enable or disable an internal BGP group.
`description` *`text`*	You can enter a brief text description of the group.
`med <0-65535>`	
`med default`	
`outdelay <0-65535>`	Specifies the amount of time in seconds that a route must be present in the routing database before it is redistributed to BGP. The configured value applies to all peers configured in this group. This feature dampens route fluctuation. Zero (0), which means that this feature is disabled. **Default:** 0
`outdelay off`	Disables outdelay.
`nexthop-self <on | off>`	Specifies for this router to send one of its own IP addresses as the BGP next hop. **Default:** off
`local-address` *`ip_address`* `<on | off>`	Specifies the IP address used on the local end of the TCP connection with the peer. Apeer session is maintained when any interface with the specified local address is operating.
`virtual-address <on | off>`	Specifies for this router to use the VRRP virtual IP address as the local endpoint for TCP connections. You must also configure a local address to enable this option. See the command above. You can configure this option only on a VRRP master. **Default**: off.
`interface [all | if_name] <on | off>`	Specifies whether to enable the specified internal peer group on all interfaces or a specific interface.

`protocol [all` *`bgp_internal_protocol`*`] <on | off>`	Specifies whether to enable all internal routing protocols on the specified internal peer group or specific internal protocols. You can enter the following specific internal protocols: *`direct`*, *`rip`*, *`static`*, *`ospf`*, and *`ospfase`*.
`peer ip_address peer_type` `<on | off>`	Specifies an internal peer address and peer type. Enter *`reflector-client`* to specify that the local router acts as a route reflector for the group of peers named. That is, the local router is the route reflection server, and the named peers are route reflection clients. Normally, the routing daemon readvertises, or reflect, routes it receives from one of its clients to all other IBGP peers, including the other peers in that client's group. Enter no-client-reflector to specify that a reflection client's routes are reflected only to internal BGP peers in other groups. Clients in the group are assumed to be direct IBGP peers of each other. Enter *`none`* if you do not want to specify route reflection.
`peer_ip_address` `weight <0-65535>`	Specifies the weight associated with the specified peer. BGP implicitly stores any rejected routes by not mentioning them in a route filter. BGP explicitly mentions them within the routing table by using a restrict keyword with a negative weight. A negative weight prevents a route from becoming active, which prevents it from being installed in the forwarding table or exported to other protocols. This eliminates the need to break and re-establish a session upon reconfiguration if import route policy is changed.
`peer` *`ip_address`* `weight off`	Disables the weight associated with the specified peer.

`peer ip_address aggregator id <on | off>`	Specifies the router's aggregate attribute as zero (rather than the router ID value). This option prevents different routers in an AS from creating aggregate routes with different AS paths **Default:** off
`peer ip_address holdtime <6-65535>`	Specifies the BGP holdtime interval, in seconds, when negotiating a connection with the specified peer. If the BGP speaker does not receive a keepalive update or notification message from its peer within the period specified in the holdtime field of the BGP open message, the BGP connection is closed.
`peer ip_address holdtime default`	Specifies a holdtime of 180 seconds.
`peer ip_address keepalive <2-21845>`	The keepalive option is an alternative way to specify a holdtime value in seconds when negotiating a connection with the specified peer. You can use the keepalive interval instead of the holdtime interval. You can also use both interval, but the holdtime value must be 3 times the keepalive interval value.
`peer ip_address_keepalive default`	Specifies a keepalive interval of 60 seconds.
`peer ip_address ignore-first-ashop <on | off>`	Specifies to ignore the first autonomous system number in the autonomous system path for routes learned from the corresponding peer. Set this option only if you are peering with a route server in transparent mode, that is, when the route server is configured to redistribute routes from multiple other autonomous systems without prepending its own autonomous system number.

`peer` *`ip_address`* `send-keepalives <on | off>`	Specifies for this router always to send keepalive messages even when an update message is sufficient. This option allows interoperability with routers that do not strictly adhere to protocol specifications regarding update.
`send-route-refresh [request | route-update] [ipv4 | ipv6 | All] [unicast]`	Specifies that the router dynamically request BGP route updates from peers or respond to requests for BGP route updates.
`peer` *`ip_address`* `accept-routes all`	Specifies an inbound BGP policy route if one is not already configured. Enter *`all`* to specify accepting routes and installing them with an invalid preference. Depending on the local inbound route policy, these routes are then made active or inactive.
`peer` *`ip_address`* `accept-routes none`	Specifies an inbound BGP policy route if one is not already configured. Enter *`none`* to specify deleting routes learned from a peer. This option saves memory overhead when many routes are rejected because no inbound policy exists.
`peer` *`ip_address`* `passive-tcp <on | off>`	Specifies for the router to wait for the specified peer to issue an open message. No tcp connections are initiated by the router. **Default:** off
`peer` *`ip_address`* `authtype none`	Specifies not to use an authentication scheme between peers. Using an authentication scheme guarantees that routing information is accepted only from trusted peers.
`peer` *`ip_address`* `authtype md5 secret` *`secret`*	Specifies to use md5 authentication between peers. In general, peers must agree on the authentication configuration to and from peer adjacencies. Using an authentication scheme guarantees that routing information is accepted only from trusted peers.

`peer` *`ip_address`* `throttle-count <0-65535>`	Specifies the number of BGP updates to send at one time. The throttle count option limits the number of BGP updates when there are many BGP peers.
`peer` *`ip_address`* `throttle count off`	Disables the throttle count option.
`peer` *`ip_address`* `log-state-transitions <on | off>`	Specifies for the router to log a message whenever a peer enters or leave the established state.
`peer` *`ip_address`* `log-warnings <on | off>`	Specifies for the rouer to log a message whenever a warning scenario is encountered in the codepath.
`peer` *`ip_address`* `trace` *`bgp_traceoption`* `<on | off>`	Specifies tracing options for your BGP implemenation. Log messages are saved in the var/log/isprd directory. Enter the following words to set each trace option. Enter *`packets`* to trace all BGP packets to this peer. Enter *`open`* to trace all the BGP open messages to this peer. Enter *`update`* to trace all the BGP update messages to this peer. Enter *`keepalive`* to trace all the keepalive messages to this peer. Enter *`all`* to trace all the message types. Enter *`general`* to trace message related to Route and Normal. Enter *`route`* to trace routing table changes for routes installed by this peer. Enter *`normal`* to trace normal protocol occurrences. Abnormal protocol occurrences are always traced. Enter *`state`* to trace state machine transitions in the protocol. Enter *`policy`* to trace application of the protocol and user-specified policy to routes being imported and exported.
`capability <default | ipv4-unicast | ipv6-unicast> <on | off>`	Specifies capabilities setting. Default is IPv4 unicast. You can set both IPv4 unicast and IPv6 unicast on.

BGP Communities

Use the following command to configure BGP communities. A BGP community is a group of destinations that share the same property. However, a community is not restricted to one network or autonomous system. Use communities to simplify the BGP inbound and route redistribution policies. Use the BGP communities commands in conjunction with inbound policy and route redistribution.

```
set bgp communities
        <on | off>
```

Arguments

`<on | off>`	Specifies whether to enable or disable BGP policy options based on communities.

BGP Show Commands

Use the following commands to monitor and troubleshoot your BGP implementation.

```
show bgp
```

```
show bgp
        groups
        memory
        errors
        paths
        stats
        peers
        peers detailed
        peer ip_address detailed
        peers established
        peer ip_address advertise
        peer ip_address received
        summary
```

OSPF

Use the following group of commands to set and view parameters for OSPF. OSPFv2 is used with IPv4 and OSPFv3 is used with IPv6. The commands for OSPFv3 are similar to those for OSPFv2, except that in place of `ospf` you enter `ipv6 ospf3`. This syntax is shown below for each set of commands and any differences in arguments used for OSPFv2 and OSPFv3 are noted in the argument tables.

> **Note**
> IPSO does not have CLI commands for route filtering and redistribution. You must configure inbound routing policies and redistribution of routes using Network Voyager. You can configure route maps and route aggregation using CLI commands. Route map configuration done through the CLI takes precedence over route filtering and redistribution configured in Voyager. For example if RIP uses route maps for inbound filtering, anything configured on the Voyager page for inbound route filters for RIP is ignored. You can still use Voyager to configure route redistribution into RIP.

When you do initial configuration, set the router ID. You can also use the following commands to change the router ID.

```
set router-id
        default
        ip_address
```

Arguments

`router-id default`	Selects the highest interface address when OSPF is enabled.
`router-id` *`ip_address`*	Specifies a specific IP address to assign as the router ID. Do not use 0.0.0.0 as the router ID address. Nokia recommends setting the router ID rather than relying on the default setting. Setting the router ID prevents the ID from changing if the default interface used for the router ID goes down.

OSPF Areas

Use the following commands to configure OSPF areas, including the backbone and stub areas.

For OSPFv2 use the following commands.

```
set ospf area
        backbone <on | off>

set ospf area ospf_area
        <on| off>
        stub <on | off>
        stub default-cost <1-677215>
        stub summary <on | off>
        nssa <on | off>
        nssa default-cost <1-677215>
        nssa default-metric-type <1-2>
        nssa import-summary-routes <on | off>
        nssa translator-role <always | candidate>
        nssa translator-stability-interval <1-65535>
        nssa redistribution <on |off>
        nssa range ip_addr [restrict] <on | off>
```

For OSPFv3 use the following commands. NSSA is not available for OSPFv3.

```
set ipv6 ospf3 area
        backbone <on | off>

set ipv6 ospf3 area ospf_area
        <on| off>
        stub <on | off>
        stub default-cost <1-677215>
        stub summary <on | off>
```

Arguments

`backbone <on | off>`	Specifies whether to enable or disable the backbone area. By default, the backbone area is enabled. You can disable the backbone area if the system does not have interfaces on the backbone area.
`<on | off>`	Specifies the area ID for a new OSPF area. Nokia recommends that you enter the area ID as a dotted quad, but you can use any integer as the area ID. The area ID 0.0.0.0 is reserved for the backbone.
`stub <on | off>`	Specifies the area ID for a stub area. Stub areas are areas that do not have AS external routes. **Note:** The backbone area cannot be a stub area.
`stub default-cost <1-677215>`	Specifies a default route into the stub area with the specified cost.
`stub summary <on | off>`	Specifies the OSPF area as totally stubby, meaning that it does not have any AS external routes and its area border routers do not advertise summary routes.
`nssa <on | off>`	Specifies the area ID for an NSSA. **Note:** The backbone area cannot be an NSSA area.
`nssa default-cost <1-677215>`	Specifies the cost associated with the default route to the NSSA.
`nssa default-metric-type <1-2>`	Specifies the type of metric. The default, type 1, is equivilent to the *Default ASE Route Type* on the OSPF Voyager page. A type 1 route is internal and its metric can be used directly by OSPF for comparision. A type 2 route is external and its metric cannot be used for comparision directly.
`nssa import-summary-routes <on | off>`	Specifies if summary routes (summary link advertisements) are imported into the NSSA.

`nssa translator-role <always | candidate>`	Specifies whether this NSSA border router will unconditionally translate Type-7 LSAs into Type-5 LSAs. When role is Always, Type-7 LSAs are translated into Type-5 LSAs regardless of the translator state of other NSSA border routers. When role is Candidate, this router participates in the translator election to determine if it will perform the translations duties.
`nssa translator-stability-interval <1-65535>`	Specifies how long in seconds this elected Type-7 translator will continue to perform its translator duties once it has determined that its translator status has been assumed by another NSSA border router. Default: 40 seconds.
`nssa redistribution <on |off>`	Specifies if both Type-5 and Type-7 LSAs or only Type-7 LSAs will be originated by this NSSA border router.
`nssa range` *`ip_addr`* `[restrict] <on | off>`	Specify the range of addresses to reduce the number of Type-5 LSAs for the NSSA border router. To prevent a specific prefix from being advertised, use the restrict argument.

OSPF Interfaces

Use the following commands to configure a backbone and other areas, such as stub areas, for specified interfaces.

For OSPFv2 use the following commands:

```
set ospf
        area <backbone | ospf_area> range ip_prefix <on | off>
        area <backbone | ospf_area> range ip_prefix restrict <on | off>
        stub-network ip_prefix <on | off>
        stub-network ip_prefix stub-network-cost <1-677722>
        interface if_name area <backbone | ospf_area> <on | off>
        interface if_name hello-interval <1-65535>
        interface if_name hello-interval default
        interface if_name dead-interval <1-65535>
        interface if_name dead-interval default
        interface if_name retransmit-interval <1-65535>
        interface if_name retransmit-interval default
        interface if_name cost <1-65535>
        interface if_name priority <0-255>
        interface if_name passive <on | off>
        interface if_name virtual-address <on | off>
        interface if_name authtype none
        interface if_name simple password
        interface if_name md5 key authorization key id secret md5 secret
        interface if_name md5 key authorization key id
```

For OSPFv3 use the same arguments as for OSPFv2 but precede them with the following:

```
set ipv6 ospf3
```

Arguments

`area <backbone | ospf_area> range ip_prefix <on | off>`	Specifies the OSPF area to which the specified interface range belongs. Select an area from the areas already configured. Any area can be configured with any number of address ranges. These ranges are used to reduce the number of routing entries that a given area transmits to other areas. If a given prefix aggregates a number of more specific prefixes within an area, you can configure an address range that becomes the only prefix advertised to other areas. Be careful when configuring an address range that covers part of a prefix that is not contained within an area. An address range is defined by an IP prefix and a mask length. If you mark a range as restrict, it is not advertised to other areas.
`area <backbone | ospf_area> range ip_prefix restrict <on | off>`	Any area can be configured with any number of address ranges. These ranges are used to reduce the number of routing entries that a given area transmits to other areas. If a given prefix aggregates a number of more specific prefixes within an area, you can configure an address range that becomes the only prefix advertised to other areas. Be careful when configuring an address range that covers part of a prefix that is not contained within an area. An address range is defined by an IP prefix and a mask length. If you mark a range as restrict, it is not advertised to other areas.

`stub-network` *`ip_prefix`* `<on | off>`	Specifies a stub network to which the specified interface range belongs. Configure a stub network to advertise reachability to prefixes that are not running OSPF. The advertised prefix appears as an OSPF internal route and is filtered at area borders with the OSPF area ranges. The prefix must be directly reachable on the router where the stub network is configured, that is, one of the router's interface addresses must fall within the prefix range to be included in the router-link-state advertisement. Use a mask length of 32 to configure the stub host. The local address of a point-to-point interface can activate the advertised prefix and mask. To advertise reachability to such an address, enter an IP address for the prefix and a non-zero cost for the prefix.
`stub-network` *`ip_prefix`* `stub-network-cost <1-677722>`	Configure a stub network to advertise reachability to prefixes that are not running OSPF. The advertised prefix appears as an OSPF internal route and is filtered at area borders with the OSPF area ranges. The prefix must be directly reachable on the router where the stub network is configured, that is, one of the router's interface addresses must fall within the prefix range to be included in the router-link-state advertisement. Use a mask length of 32 to configure the stub host. The local address of a point-to-point interface can activate the advertised prefix and mask. To advertise reachability to such an address, enter an IP address for the prefix and a non-zero cost for the prefix.
`interface` *`if_name`* `area <backbone | ospf area> <on | off>`	Specifies the OSPF area to which the specified interface belongs.

`interface` *if_name* `hello-interval <1-65535>`	Specifies the interval, in seconds, between hello packets that the router sends on the specified interface. For a given link, this value must be the same on all routers or adjacencies do not form.
`interface` *if_name* `hello-interval default`	Specifies the default value for the hello interval, which is 10 seconds.
`interface` *if_name* `dead-interval <1-65535>`	Specifies the number of seconds after which a router stops receiving hello packets that it declares the peer down. Generally, you should set this value at 4 times the value of the hello interval. Do not set the value at 0. For a given link, this value must be the same on all routers or adjacencies do not form.
`interface` *if_name* `dead-interval default`	Specifies the default value for the dead interval, which is 40 seconds
`interface` *if_name* `retransmit-interval <1-65535>`	Specifies the number of seconds between link state advertisement transmissions for adjacencies belonging to the specified interface. This value also applies to database description and link state request packets. Set this value conservatively, that is, at a significantly higher value than the expected round-trip delay between any two routers ont he attached network.
`interface` *if_name* `retransmit-interval default`	Specifies the default default for the retransmit interval, which is 5 seconds.
`interface` *if_name* `cost <1-65535>`	Specifies the weight of the given path in a route. The higher the cost, the less preferred the link. To use one interface over another for routing paths, assign one a higher costo.

`interface if_name priority <0-255>`	Specfies the priority for becoming the designated router (DR) on the specified link. When two routers attached to a network attempt to become a designated router, the one with the highest priority wins. This option prefents the DR from changing too often. The DR option applies only to a share-media interface, such as Ethernet or FDDI; a DR is not elected on a point-to-point type interface. A router with a priority of 0 is not eligible to become the DR.
`interface if_name passive <on | off>`	Enabling this option puts the specified interface into passive mode; that is, hello packets are **not** sent from the interface. Putting an interface into passive mode means that no adjacencies are formed on the link. This mode enables the network associated with the specified interface to be included in intra-area route calculation rather than redistributing the network into OSPF and having it function as an autonomous system external. **Default:** off
`interface if_name virtual-address <on | off>`	Enables OSPF on the virtual IP address associated with this interface.This option functions only if this router is a VRRP master. You must also configure VRRP to accept connections to VRRP IPs. See "General VRRP Commands" for more information. **Default**: off **Note** You must use Monitored Circuit mode when configuring virtual IP support for OSPF or any other dynamic routing protocol. Do not use VRRPv2 when configuring virtual IP support for OSPF.
`interface if_name authtype none`	Specifies not to use an authentication scheme for the specified interface.

`interface` *`if_name`* `authtype simple` *`password`*	Specifies to use simple authentication for the specified interface. Enter an ASCII string that is 8 characters long. Generally, routers on a given link must agree on the authentication configuration to form peer adjacencies. Use an authentication scheme to guarantee that routing information is accepted only from trusted peers.
`interface` *`if_name`* `authtype md5 key` *`authorization key id`* `secret` *`md5 secret`*	Specifies to use MD5 authorization. Enter at least one key ID and its corresponding MD5 secret. If you configure multiple key IDs, the largest key ID is used for authenticating outgoing packets. All keys can be used to authenticate incoming packets. Generally, routers on a given link must agree on the authentication configuration to form peer adjacencies. Use an authentication scheme to guarantee that routing information is accepted only from trusted peers.

OSPF Virtual Links

Use the following commands to configure OSPF virtual links. Configure a virtual link if the router is a border router that does not have interfaces in the backbone area. The virtual link is effectively a tunnel across an adjacent non-backbone area whose endpoint must be any of the adjacent area's border routers that has an interface in the backbone area.

For OSPFv2 use the following commands:

```
set ospf area backbone virtual-link
        ip_address transit-area ospf_area <on | off>
        ip_address transit-area ospf_area hello-interval <1-65535>
        ip_address transit-area ospf_area hello-interval default
        ip_address transit-area ospf_area dead interval <1-4294967295>
        ip_address transit-area ospf_area dead interval default
        ip_address transit-area ospf_area retransmit-interval
          <1-4294967295>
        ip_address transit-area ospf_area retransmit-interval default
        ip_address transit-area ospf_area authtype none
        ip_address transit-area ospf_area authtype simple password
        ip_address transit-area ospf_area authtype md5 key
          authorization key id secret md5 key
        ip_address transit-area ospf_area authtype md5 key authorization
          key id off
```

For OSPFv3 use the following with the same arguments as for OSPFv2:

```
set ipv6 ospf3 area backbone virtual-link
```

Arguments

ip_address transit-area *ospf_area* <on | off>	Specifies the IP address of the remote endpoint of the virtual link and transit area, which is a specified ospf area you configure using the set ospf area command. Configure the ospf area you are using as the transit area before you configure the virual link. The transit area is the area shared by the border router on which you configure the virtual link and the router with an interface in the backbone area. Traffic between the endpoints of the virtual link flow through this area. The virtual link IP address functions as the router ID of the remote endpoint of the virtual link.

ip_address `transit-area` *ospf_area* `hello-interval <1-65535>`	Specifies the interval, in seconds, between hello packets that the router sends on the specified interface. For a given link, this value must be the same on all routers or adjacencies do not form.
ip_address `transit-area` *ospf_area* `hello-interval default`	Specifies an interval of 10 seconds.
ip_address `transit-area` *ospf_area* `dead-interval <1-4294967295>`	Specifies the number of seconds after which a router stops receiving hello packets that it declares the neighbor down. Generally, you should set this value at 4 times the value of the hello interval. Do not set the value at 0. For a given link, this value must be the same on all routers or adjacencies do not form.
ip_address `transit-area` *ospf_area* `dead-interval default`	Specifies a value of 40 seconds.
ip_address `transit-area` *ospf_area* `retransmit-interval <1-4294967295>`	Specifies the number of seconds between link state advertisement transmissions for adjacencies belonging to the specified interface. This value also applies to database description and link state request packets. Set this value conservatively, that is, at a significantly higher value than the expected round-trip delay between any two routers ont he attached network.
ip_address `transit-area` *ospf_area* `retransmit-interval default`	Specifies a value of 5 seconds.
ip_address `transit-area` *ospf_area* `authtype none`	Specifies not to use an authentication scheme for the specified interface.

ip_address transit-area *ospf_area* authtype simple *password*	Specifies to use simple authentication for the specified interface. Enter an ASCII string that is 8 characters long. Generally, routers on a given link must agree on the authentication configuration to form neighbor adjacencies. Use an authentication scheme to guarantee that routing information is accepted only from trusted peers.
ip_address transit-area *ospf_area* authtype md5 key *authorization key id* secret *MD5 secret*	Specifies to use MD5 authorization. Enter at least one key ID and its corresponding MD5 secret. If you configure multiple key IDs, the largest key ID is used for authenticating outgoing packets. All keys can be used to authenticate incoming packets. Generally, routers on a given link must agree on the authentication configuration to form neighbor adjacencies. Use an authentication scheme to guarantee that routing information is accepted only from trusted peers.

OSPF Global Settings

Use the following commands to configure setting that apply to all configured OSPF areas, including the backbone and stub areas.

For OSPFv2 use the following commands:

```
set ospf
        rfc1583-compatibility <on | off>
        spf-delay <1-60>
        spf-delay default
        spf-holdtime <1-60>
        spf-holdtime default
        default-ase-cost <1-677215>
        default-ase-type <1 | 2>
```

For OSPFv3 use the following commands:

```
set ipv6 ospf3
        spf-delay <1-60>
        spf-delay default
        spf-holdtime <1-60>
        spf-holdtime default
        default-ase-cost <1-677215>
        default-ase-type <1 | 2>
```

Arguments

`rfc1583-compatibility <on | off>`	The Nokia implementation of OSPF is based on RFC 2178, which fixed some looping problems in an earlier specification of OSPF. If your implementation runs in an environment with OSPF implementations based on RFC 1583 or earlier, enable this option, which is on by default. Setting compatibility with RFC 1583 ensures backward compatibility. This argument is not used with OSPFv3. **Default:** on
`spf-delay <1-60>`	Specifies the time, in seconds, to wait before recalculating the OSPF routing table after a change in the topology.
`spf-delay default`	Specifies an spf-delay time of 2 seconds.
`spf-holdtime <1-60>`	Specifies the minimum time, in seconds, between recalculations of the OSPF routing table.
`spf-holdtime default`	Specifies an spf-holdtime of 5 seconds.
`default-ase-cost <1-6777215>`	Specifies the cost assigned to routes from other protocols that are redistributed into OSPF as autonomous systems external. If the route has a cost already specified, that cost takes precedent. **Default:** 1

`default-ase-type <1 | 2>`	Specifies the type assigned to routes from other protocols that are redistributed into OSPF as autonomous systems external. If the route has a type already specified, that type takes precedent. **Default:** 1

OSPF Show Commands

Use the following commands to monitor and troubleshoot your OSPF implementation.

To view a summary of your OSPF implementation, including the number of areas configured and the number of interfaces configured within each area, use `show ospf` (for OSPFv2) or `show ipv6 ospf3` (for OSPFv3).

For OSPFv2 use the following commands:

```
show ospf
        neighbors
        neighbor ip_address
        interfaces
        interfaces stats
        interfaces detailed
        interface ifname
        interface ifname stats
        interface ifname detailed
        packets
        errors
        errors dd
        errors hello
        errors ip
        errors lsack
        errors lsr
        errors lsu
        errors protocol
        events
        border-routers
        database
        database areas
        database area ospf_area
        database asbr-summary-lsa
        database checksum
        database database-summary
        database detailed
        database external-lsa
        database network-lsa
        database router-lsa
        database summary-lsa
        database type <1 | 2 | 3 | 4 |5 | 7> [detailed]
        database nssa-external-lsa [detailed]
        summary
```

For OSPFv3 use the following commands:

```
show ipv6 ospf3
        neighbors
        neighbor ip_address
        interfaces
        interfaces stats
        interfaces detailed
        interface ifname
        interface ifname stats
        interface ifname detailed
        packets
        errors
        errors dd
        errors hello
        errors ip
        errors lsack
        errors lsr
        errors lsu
        errors protocol
        events
        border-routers
        database
        database areas
        database area ospf area
        database checksum
        database database-summary
        database detailed
        database external-lsa
        database inter-area-prefix
        database inter-area-router-lsa
        database intra-area-prefix-lsa
        database link-lsa
        database network-lsa
        database router-lsa
        database type <1-5>
        database events
        summary
```

Arguments

`neighbors`	Displays the IP addresses of neighboring interfaces, their priority and status, and the number of errors logged for each interface.
`neighbor` *`ip_address`*	Displays the priority, status, and number of errors logged for the specified IP address.
`interfaces`	Displays the names of all configured logical interfaces, their corresponding IP addresses, to area to which each interface is assigned, each interface's status and the IP addresses of each logical interface's designated router and backup designated router.
`interfaces stats`	Displays the number of each type of error message logged for each OSPF interface as well as the number of link state advertisements sent by each interface.
`interfaces detailed`	Displays detailed information about each OSPF interface, including the authentication type configured if any, the router IDs and IP addresses of the designated router and backup designated router, the timer intervals configured for hello wait, dead, and restransmit messages, and the number of neighbors for each interface.
`interface` *`if_name`*	Displays the IP address, area ID, status, number of errors logged, and the IP address of the designated router and backup designated router for the specified itnerface.
`interface` *`if_name`* `stats`	Displays the number of each type of error message logged by the specified interface as well as the number of link-state advertisements sent by the specified interface.

`interface` *`if_name`* `detailed`	Displays detailed information about the specified interface, including the authentication type configured if any, the router IDs and IP addresses of the designated router and backup designated router, the timer intervals configured for hello wait, dead, and restransmit messages, and the number of neighbors for each interface
`packets`	Diplays the number of each type of packet sent, including hello packets, link-state update packets, and link-state acknowledgment and link-state request packets.
`errors`	Dipslays the number of each type of error message sent, including hello protocol errors, database description errors, protocol errors, link-state acknowledgment errors, link-state request errors, link-state update errors, and IP errors.
`errors dd`	Displays the number of each type of database- description error messages only.
`errors hello`	Displays the number of each type of hello- error message only.
`errors ip`	Displays the number of each type of IP-errors message only.
`errors lsack`	Displays the number of each type of link-state acknowledgment error message only.
`errors lsu`	Displays the numberof each type of link-state update error message only
`errors lsr`	Displays the number of each type of link-state request error messages only.
`errors protocol`	Displays the number of each type of protocol error message only.
`border-routers`	Displays the IP address of each area border router, the OSPF area of each border router, and the cost associated with each IP address.

`database`	Displays router-link state and network-link sate statistics for each OSPF area. Also displays the checksum, sequence number, and link count of each OSPF interface.
`database areas`	Displays router-link state, network-link state, AS-border-router link state, AS-external link state, and summary-link state statistics for each OSPF area. Also displays the checksum, sequence number, and link count of each OSPF interface.
`database area` *`ospf_area`*	Displays router-link state, network-link state, AS-border-router-link state, AS- external-link state, and summary-link state statistics for the speciified OSPF area. Also displays the checksum, sequence number, and link count of each IP address configured within the specified OSPF area.
`database asbr-summary`	Displays a summary of AS-border-router link state statstics for each OSPF area. For OSPFv2 only.
`database inter-area-router-lsa`	Displays a summary of AS-border-router link state statstics for each OSPF area. For OSPFv3 only.
`database external`	Displays AS-external-link state statistics for each OSPF area.
`database database-summary`	Displays a summary of router-link-state, network-link state, summary-link-state , and AS-border-router-link state statistics.
`database network`	Displays network-link-state statistics, including the advertised router, sequence number, and checksum of each OSPF interface. For OSPFv2 only.
`database nssa-external-lsa [detailed]`	Displays type 7 LSAs (NSSA). This argument applies only to OSPF v2; OSPFv3 is not NSSA aware.
`database router-lsa`	Displays router-link-state statistics, including the advertised router, sequence number, checksum, an link count, of each OSPF interface. For OSPFv2 only.

`database summary-lsa`	Displays a summary of link-state statistics for each OSPF area. For OSPFv2 only.
`database inter-area-prefix-lsa`	Displays a Type 3 summary of link-state statistics for each OSPF area. For OSPFv3 only.
`database intra-area-prefix-lsa`	In OPSFv3 all addressing information is removed from router lsa and network lsa and intra-area-prefix-lsa carries this addressing information. It associates a list of IPv6 address prefixes with a transit network link by referencing a network lsa or a router lsa. A stub link's prefixes are carried by an intra-area-prefix lsa that references a router-lsa. For OSPFv3 only.
`link lsa`	Describes a router's link-local address and the IPv6 address prefixes associated with a link. For OSPFv3 only.
`database type <1 | 2 | 3 | 4 |5 | 7> [detailed]`	Displays link-state statistics associated with the specified number: • 1—router-link-state statistics. • 2—network-link-state statistics. • 3—summary-link-state statistics. • 4—AS-border-router-link-state statistics. • 5—AS-external-link-state statistics. • 7—NSSA. This option applies only to OSPF v2; OSPFv3 is not NSSA aware.
`events`	Displays the number of interface up/down events; virtual interface up/down events; designated router election events; router ID changes; area border router changes; AS border router changes, and link state advertisement messages.

RIP

Use this group of commands to set and view parameters for RIP.

Note
IPSO does not have CLI commands for route filtering and redistribution. You must configure inbound routing policies and redistribution of routes through Voyager. You can configure route maps and route aggregation using CLI commands. Route map configuration done through the CLI takes precedence over route filtering and redistribution configured in Voyager. For example if RIP uses route maps for inbound filtering, anything configured on the Voyager page for inbound route filters for RIP is ignored. You can still use Voyager to configure route redistribution into RIP.

Use these commands to configure RIP properties for specific interfaces.

RIP Interfaces

```
set rip interface if_name
        off
        version <1 | 2> on
        metric <0-16>
        metric default
        accept-updates <on | off>
        send-updates <on | off>
        transport multicast
        transport broadcast
        authtype none
        authtype simple password
        authtype md5 secret secret [cisco-compatibility] <on | off>
        virtual address <on | off>
```

General RIP Properties

Use these commands to configure RIP properties that apply to all interfaces configured for RIP.

```
set rip
        auto-summary <on | off>
        update-interval <1-65535>
        update-interval default
        expire-interval <1-65535>
        expire-interval default
```

Arguments

`<1 | 2>`	Specifies the version of RIP to run.
`metric <0-16>`	Specifies the rip metric added to routes set that use the specified interface.
`metric default`	Specifies a value of 0.
`accept-updates <on | off>`	Specifies whether to accept RIP packets using the specified interface. **Default:** off
`send-updates <on | off>`	Specifies whether RIP packets should be sent using the specified interface.
`transport multicast`	Specifies for RIP version 2 packets to be multicast on the specified interface.
`transport broadcast`	Specifies for RIP version 1 packets that are compatible with rip version 2 to be broadcast on the specified interface.
`authtype none`	Specifies not to implement an authentication scheme for the specificed interface to accept routing information from neighboring routers. This option applies to rip version 2 only.

`authtype simple` *`password`*	Specifies to implement a simple authentication sheme for the specified interface to accept routing information from neighboring routers. The password must contain alphanumeric characters only and can be between one and 16 characters long. This option applies to RIP version 2 only
`authtype md5 secret` *`secret`*	Specifies to implement an authentication scheme that uses an MD5 algorithm for the specified interface to accept routing information from neighboring routers. This option applies to RIP version 2 only.
`interface` *`if_name`* `virtual <on | off>`	Enables RIP on the virtual IP address associated with this interface.This option functions only if this router is a VRRP master. You must also configure VRRP to accept connections to VRRP IPs. See "ICMP Router Discovery" for more information. **Default**: off **Note**: You must use Monitored Circuit mode when configuring virtual IP support for any dynamic routing protocol, including RIP. Do not use VRRPv2 when configuring virtual IP support for RIP or any dynamic routing protocol.
`cisco-compatibility <on | off>`	Specifies whether to interoperate with Cisco routers also using the MD5 authenitcation scheme. **Default:** off
`auto-summary <on | off>`	Specifies whether to aggregate and distribute non-classful routes when using RIP version 1. **Default:** on
`update-interval <1-65535>`	Specifies the amount of time, in seconds, between RIP updates.
`update-interval default`	Specifies a value of 30 seconds.

`expire-interval <1-65535>`	Specifies the amount of time, in seconds, that must pass without receiving an update for a given route before the route is considered to have timed out. This value should be 6 times the update interval in order to allow for the possibility that packets containing an update could be dropped bythe network.
`expire-interval default`	Specifies a value of 180 seconds.

RIP Show Commands

Use these commands to monitor and troubleshoot RIP.

```
show rip
        interfaces
        interface <if_name>
        packets
        errors
        neighbors
        summary
```

IGRP

Use these commands to set and view parameters for the Interior Gateway Routing Protocol.

> **Note**
> IPSO does not have CLI commands for route filtering and redistribution. You must configure inbound routing policies and redistribution of routes through Voyager. You can configure route maps and route aggregation using CLI commands. Route map configuration done through the CLI takes precedence over route filtering and redistribution configured in Voyager. For example if RIP uses route maps for inbound filtering, anything configured on the Voyager page for inbound route filters for RIP is ignored. You can still use Voyager to configure route redistribution into RIP.

General IGRP Properties

```
set igrp
        as <0-65535>
        as off
        default-delay <0-16777215>
        default-delay off
        default-bandwidth <1-677215>
        default-bandwidth off
        default-reliability <0-255>
        default-reliability off
        default-load <1-255>
        default-load off
        default-mtu <1-65535>
        default-mtu off
        k1 <0-16777215>
        k1 default
        k2 <0-16777215>
        k2 default
        holddown <on | off>
        max-hop-count <1-255>
        max-hop-count default
        update-interval <1-65535>
        update-interval default
        invalid-interval <1-65535>
        invalid-interval default
        hold-interval <1-65535>
        hold-interval default
        flush-interval <1-65535>
        flush-interval default
        validate fields <on | off>
```

IGRP Interfaces

Use these commands to configure IGRP properties for specific interfaces.

```
set igrp interface if_name
        <on | off>
        delay <1-16777215>
        bandwidth <1-6777215>
        accept-updates <on | off>
```

Arguments

`as <0-65535>`	Specifies the autonomous system number of IGRP packets. You do not have to use an officially registered as number, but if your organization has one, you should use that number. Update messages also include the as number.
`as off`	Specifies to disable the autonomous system number. Because you must enable an autonomous system number to run IGRP, disabling the as and not configuring a new as means that you cannot run IGRP.
`default-delay <0-16777215>`	Specifies IGRP delay metrics in units of 10 microseconds. Set this option if you are exporting routes into IGRP.
`default-bandwidth <1-16777215>`	Specifies the IGRP bandwidth metric in units of inverted bits/second scaled by a factor of 10,000,000,000. Set this option if you are exporting routes into IGRP.
`default-reliability <0-255>`	Specifies the IGRP reliability metric as a fraction of 255, that is, 255=100%. Set this option if you are exporting routes into IGRP.
`default-load <1-255>`	Specifies the IGRP load metric as a fraction of 255, that is, 255=100%. Set this option if you are exporting routes into IGRP.
`default-mtu <0-65535>`	Specifies the IGRP maximum transmission unit. Set this option if you are exporting routes into IGRP.
`k1 <0-1677215>`	Specifies the IGRP bandwidth multiplier constant used in the composite metric computation.

`k1 default`	Specifies a value of 1.
`k2 <0-1677215>`	Specifies the IGRP delay multiplier constant used in the composite metric computation.
`k2 default`	Specifies a value of 1.
`holddown <on | off>`	Specifies whether IGRP performs "holddown loop" prevention measures. This setting should be consistent throughout an autonomous system. Enabling holddowns has the effect of disabling the stronger form of route poisoning. **Default:** on
`max-hop-count <1-255>`	Specifies the maximum allowable "hop count" an incoming route must have in order to be accepted. For a route to be marked as "reachable" in an update, its “hop count” must not exceed this value.
`max-hop-count default`	Specifies a value of 100.
`update-interval <1-65535>`	Specifies the amount of time, in seconds, between regularly scheduled updates
`update-interval default`	Specifies a value of 90.
`invalid-interval <1-65535>`	Specifies the amount of time, in seconds, that must pass without receiving an update for a given route before the route is considered to have timed out
`invalid-interval default`	Specifies a value of 3 times the update interval value.

`hold-interval <1-65535>`	Specifies the amount of time, in seconds, that a route remains in a "hold down" state. The interval should be several times the value of the update interval. The hold interval must be at least as long as the flush interval minus the value of the invalid interval. When a route has become unreachable (or the metric has increased enough to cause poisoning), the route goes into a "hold down" state (when the Holddown field is enabled). During this state, no new route is accepted for the same destination for this amount of time.
`hold-interval default`	Specifies a value that is 3 times the configured update interval value plus 10.
`flush-interval <1-65535>`	Specifies the amount of time, in seconds, before a routing table entry is removed. The interval should be longer than the sum of the invalid interval and the hold interval values. After the Invalid interval expires, a route is timed out and marked "unreachable". The routing table entry for the destination remains, in order to enforce the holddown.
`flush-interval default`	Specifies a value 7 times the configured update interval value.
`validate-fields <on | off>`	Specifies that IGRP should not check that reserved fields are zero in incoming IGRP request packets. Normally, IGRP rejects request packets when the reserved fields are not zero. The reserved fields in a request packet are the "edition" number and the three "route counts". When you enable this option, any possible trailing data after the IGRP header is ignored. Normally, IGRP rejects request packets that are not exactly the size of the IGRP header.
`interface` *`if_name`* `<on | off>`	Specifies whether to enable or disable IGRP on the specified interface.

`interface` *`if_name`* `delay <1-16777215>`	Specifies the IGRP delay metric in units of 10 microseconds.
`interface` *`if_name`* `bandwidth <1-1677215>`	Specifies the IGRP bandwidth metric in units of inverted bits/second scaled by a factor of 10,000,000.
`interface` *`if_name`* `accept-updates <on | off>`	Specifies whether IGRP packets received through the specified interface are accepted or ignored. **Default:** on

IGRP Show Commands

Use these commands to monitor and troubleshoot IGRP.

```
show igrp
        errors
        interfaces
        interface if_address
        neighbors
        packets
        policy
        route-stats
```

IGMP

Use this group of commands to configure parameters for the internet group management protocol.

IGMP Commands

Use these commands to configure IGMP for specific interfaces.

```
set igmp interface if_name
        last-member-query-interval <1-25>
        last-member=query-interval default
        local-group address <on | off>
        loss-robustness <1-255>
        loss-robustness default
        query-interval <1-3600>
        query-interval default
        query-response-interval <1-25>
        query-response-interval default
        router-alert <on | off>
        static-group address <on | off>
        version <1 | 2 | 3>
```

Use the following commands when IP clustering is enabled. You must be logged in as a cluster administrator. These commands are not functional unless IP clustering is enabled.

```
set igmp network ip_address/mask length
        last-member-query-interval <1-25>
        last-member=query-interval default
        local-group address <on | off>
        loss-robustness <1-255>
        loss-robustness default
        query-interval <1-3600>
        query-interval default
        query-response-interval <1-25>
        query-response-interval default
        router-alert <on | off>
        static-group address <on | off>
        version <1 | 2 | 3>
```

Arguments.

`network` *`ip_address/mask length`*	Specifies the cluster network on which IGMP should be enabled.
`last-member-query-interval <1-25>`	Specifies the maximum response time, in seconds, inserted into IGMP group-specific queries.
`last-member-query-interval default`	Specifies a value of 1.
`local-group` *`address`* `<on | off>`	Specifies a multicast group address. IPSO acts as a receiver for this group and buildd the reverse path forwarding tree without waiting for requests from downstream hosts. IGMP informs the parent multicast protocol about the simulated local receiver and sends a membership report out of this interface.
`loss-robustness <1-255>`	Specifies a value that corresponds to the expected packet loss on a subnet.
`loss-robustness default`	Specifies a value of 2.
`query-interval <1-3600>`	Specifies the interval, in seconds, between IGMP general queries.

`query-interval default`	Specifies a value of 125.
`query-response-interval <1-25>`	Specifies the maximum response time, in seconds, inserted into the periodic IGMP general queries
`query-response-interval default`	Specifies a value of 10.
`router-alert <on | off>`	Specifies that the router-alert option not be set in IGMP messages sent on this interface. **Default:** off
`static-group` *`address`* `<on | off>`	Specifies a multicast group address. IPSO acts as a receiver for this group and buildd the reverse path forwarding tree without waiting for requests from downstream hosts. IGMP informs the parent multicast protocol about the simulated local receiver but does not send a membership report out of this interface.
`version <1 | 2 | 3>`	Specifes which version of IGMP to run. IGMP version 2 is compatible with IGMP version 1, and version 3 is compatible with versions 2 and 1. Nokia recommends that you use version 1 only on networks that include multicast routers that are not upgraded to IGMP versions 2 or 3.

IGMP Show Commands

Use these commands to monitor and troubleshoot IGMP.

```
show igmp
        stats
        stats receive
        stats transmit
        stats error
        interfaces
        interfaces if_address
        groups [local | static [interface logical_interface]]
        group if_address
        if-stats
        if-stat if_address
        summary
```

Use the following commands to monitor and troubleshoot IGMP when IP clustering is enabled.

```
show igmp
        networks
        network ip_address/mask length
        show igmp net-stats
        show igmp net-stat ip_address/masklength
        stats [receive | transmit | summary]
        summary
```

PIM

Use this group of commands to configure parameters for PIM.

```
set pim mode
        <dense | sparse>
```

PIM Interfaces

After you set PIM to run either dense or sparse mode, use the following commands to configure PIM for specific interfaces.

```
set pim interface if_name
        <on | off>
        virtual-address <on | off>
        local-address ip_address
        dr-priority <0-4294967295>
        dr-priority default
```

PIM With IP Clustering

To use the following commands, you must be logged in as cadmin. These commands are not available unless you are logged in as cadmin. Any configuration you perform when logged in as cadim is automatically propogated to each node of the cluster. When a new node joins a cluster, the local configuration of that node is replaced by the configuration obtained from the master.

```
set pim network ip_address/mask length
        <on | off>
        dr-priority <0-4294967295>
        dr-priority default
```

Sparse Mode PIM

Use the following commands to configure parameters for sparse mode PIM only.

```
set pim
        ha-mode <on | off>
        bootstrap-candidate <on | off>
        bootstrap-candidate local-address ip_address
        bootstrap-candidate priority <0-255>
        bootstrap-candidate priority default
        candidate-rp <on | off>
        candidate-rp local-address ip_address
        candidate-rp priority <0-255>
        candidate-rp priority default
        candidate-rp multicast group mcast_ip_prefix <on | off>
        static-rp off
        static-rp rp-address ip_iddresss < on | off>
        static-rp rp-address ip_address multicast-group mcast_ip_prefix
          <on | off>
        register-suppress-interval <60-3600>
        register-suppress-interval default
        candidate-rp advertise-interval <1-3600>
        candidate rp-advertise-interval default
        cisco compatibility <on | off>
        spt-threshold multicast mcast_ip_prefix threshold <0-1000000>
          <on | off>
        spt-threshold multicast mcast_ip_prefix threshold infinity
          <on | off>
```

Timer and Assert Rank Parameters for Dense Mode and Sparse Mode

Use these commands to change or restore default values for timers and assert ranks.

```
set pim
        hello-interval <1-21845>
        hello-interval default
        data-interval <11-3600>
        data-interval default
        assert-interval <1-3600>
        assert-interval default
        assert-limit <10-10000>
        assert-limit default
        assert-limit <0>
        jp-interval <1-3600>
        jp-interval default
        jp-delay-interval <1-3600>
        jp-delay-interval default
        jp-suppress-interval <2-3600>
        jp-suppress-interval default
        assert-rank protocol protocol name rank <0-255>
        assert-rank protocol protocol name rank default
```

Arguments

<dense | sparse>	Specifies whether to run PIM dense sparse mode.
interface *if_name* <on | off>	Specifies whether to enable or disable PIM on a specified interface.

`virtual-address <on | off>`	Specifies to enable VRRP virtual IP address on the specified PIM interface. This option lets you configure a etiher a PIM Sprase-Mode or PIM Dense-Mode interface to advertise the VRRP virtual IP address if the router transitions to become VRRP master after a failover. When you enable virtual IP support for VRRP on a PIM interface, it establishes the neighbor relationship using the virtual IP if the router is a VRRP master. The master in the VRRP pair sends hello messages that include the virtual IP as the source address and processes PIM control messages from routers that neighbor the VRRP pair. **Note**: You must use Monitored Circuit mode when configuring virtual IP support for any dynamic routing protocol, including PIM, either sparse-mode or dense-mode. Do not use VRRPv2 when configuring virtual IP support for any dynamic routing protocol.
`local-address` *`ip_address`*	Specifies the local address used in all advertisements sent on the interface. This option is useful when multiple multiple IP addresses are configured on the interface If you enter an address other than one configured for that interface, PIM ignores your configured address and selects one of the addresses configured on the interface. Warning: If neighboring routers choose advertisement addresses that do not appear to be on a shared subnet, all messages from the neighbor will be rejected. Thus, a PIM router on a shared LAN must have at least one interface address with a subnet prefix shared by all neighboring PIM routers.

`ha-mode <on | off>`	Specifies whether to enable or disable the High Availability (HA) mode. Enable the High-Availability (HA) mode when two routers are configured to back each other up to forward multicast traffic and sparse-mode PIM is implemented. When this option is enabled, all PIM-enabled interfaces are available only if each interface is up and has a valid address assigned. If any PIM-enabled interface goes down or all its valid addresses are deleted, then all PIM-enabled interfaces become unavailable and remain in that state until all interfaces are back up. • The HA mode feature applies only to sparse-mode PIM. The HA mode feature does not affect the functioning of dense-mode PIM. **Note**: Beginning with IPSO 3.8, you can configure PIM to advertise the virtual VRRP IP address on a interface with PIM enabled. You do not need to enable HA mode if you configure the interface to advertise the virtual VRRP IP address. **Default:** off
`dr-priority <0-4294967295>`	Specifies the dr-priority advertised in the PIM hello messages sent on the corresponding interface. This value, which has a default of 1, is used for DR election on a LAN. The router with the highest priority and the highest IP address is elected the designated router. To break a tie, the DR is selected on the basis of the highest IP address. If even one router does not advertise a dr-priority value in its hello messages, the DR election is baed on the IP address.
`dr-priority default`	Specifies a value of 1.

`bootstrap-candidate <on | off>`	Specifies that the platform is a candidate bootstrap router. The boostrap router collects candidate rendezvous point information and disseminates rp-set information associated with each group prefix. To avoid a single point of failure, configure more than router in a domain as a candidate bootstrap router. **Default:** off
`bootstrap-candidate local-address` *`ip_address`*	Specifies the IP address of the bootstrap router used in bootstrap messages. By default, the router picks an address from one of the interfaces on which PIM is enabled.
`bootstrap-candidate priority <0-255>`	Specifies the value used to elect the boostrap router from among the candidate bootstrap routers. The candidate bootstrap router with the highest priority value is elected bootstrap router for the domain. The highest priority value is 0, so the lower the value, the higher the priority.
`bootstrap-candidate priority default`	Specifies a value of 0.
`candidate-rp <on | off>`	Specifies that the platform is a candidate rendezvous point router. **Default:** off
`candidate-rp local-address` *`ip_address`*	Specifies the IP address of the candidate rendezvous point router used in candidate rendezvous point messages. By default, the router picks an address from one of the interfaces on which PIM is enabled.
`candidate-rp priority <0-255>`	Specifies the priority of the candidate rendezvous point included in the corresponding multicast group address. The higher the priority, the lower the value.
`candidate-rp priority default`	Specifies a value of 0.

`candidate-rp multicast-group` *`mcast_ip_prefix`* `<on | off>`	Specifies the multicast address advertised in the candidate rendezvous point advertisements. For the multicast IP prefix value, you must enter an IP address and mask length. If you do not specify a group multicast address, the candidate rendezvous point advertises itself as the rendezvous point for all multicast groups.
`static-rp off`	Disables the static rendezvous point option.
`static-rp rp-address` *`ip_address`* `<on | off>`	Specifies to enable or disable a static rendezvous point. If you do not specify an associated multicast group and prefix, the static-rp is considered to be responsible for all multicast groups (224.0.0.0/4).
`static-rp rp-address` *`ip_address`* `multicast-group` *`mcast_ip_prefix`* `<on | off>`	Specifies the IP address associated with the static rendezvous point and the multicast IP address for which the rendezvous point is responsible. For the multicast IP prefix value, you must enter an IP address and mask length.
`register-suppress-interval <60-3600>`	Specifies the mean interval between receiving a register-stop and allowing registers to be sent again. A lower value means more frequent register bursts at the rendezvous point, while a higher value means a longer join latency for new receivers.
`register-suppress-interval default`	Specifies a value of 60.
`candidate-rp advertise-interval <1-3600>`	Specifies the interval between which candidate-rendezvous point routers send candidate-rendezvous point advertisements.
`candidate-rp advertise-interval default`	Specifies a value of 60.

`cisco-compatibility <on | off>`	The checksum of the PIM register messages is calculated without including the maltiest payload. Earlier releases of Cisco's IOS calculate the checksum by including the multicast payload. If you experience difficulties having PIM register messages sent by your Nokia appliance being accepted by a Cisco router that is the elected rendezvous point (RP), configure this option. A Nokia appliance that is the elected RP, accepts register messages that calculate the checksum with or without the multicast payload, that is it accepts all register messages. **Default:** off
`spt-threshold multicast` *`mcast_ip_prefix`* `threshold <0-1000000>`	Specifies the multicast group address to apply to the shortest path tree (spt) threshold and the date rate in kbits/sec to trigger the spt switch over.
`spt-threshold multicast` *`mcast_ip_prefix`* `threshold infinity <on | off>`	Specifies the data rate in kibts/sec to trigger the spt switch over as infinity.
`hello interval <1-21845>`	Specifies the interval, in seconds, at which PIM hello messages are sent on the LAN.
`hello interval default`	Specifies a value of 30.
`data-interval <11-3600>`	Specifies the interval, in seconds, after which multicast (S,G) state for a silent source is deleted.
`data-interval default`	Specifies a value of 210.
`assert-interval <1-3600>`	Specifies the interval between the last time an assert is received and the assert is timed out.
`assert-interval default`	Specifies a value of 180.
`assert-limit <10-10000>`	Specifies the number of asserts to send per second.

`assert-limit default`	Specifies a value of 10.
`assert-limit <0>`	Disables the limit placed on the number of asserts that can be sent per second.
`jp-interval <1-3600>`	Specifies the interval, in seconds, between which join/prune messages are sent.
`jp-interval default`	Specifies a value of 60.
`jp-delay-interval <1-3600>`	Specifies maximum interval, in seconds, between the time when the RPF neighbor changes and a triggered Join/Prune message is sent.
`jp-delay-interval default`	Specifies a value of 5.
`jp-suppress-interval <2-3600>`	Specifies the mean interval between receiving a Join/Prune with a higher “holdtime” and allowing duplicate Join/Prunes to be sent again. Nokia recommends that you set the join/prune suppress interval 1.25 times that of the join/prune interval.
`jp-suppress-interval default`	Specifies a value of 75.
`assert-rank protocol` *`protocol name`* `rank <0-255>`	Specifies the value assigned to a particular protocol in assert messages. This value is used to compare protocols to determine which router will forward multicast packets on a multi-access LAN. The value is included in assert messages when more than one router on a LAN is capable of forwarding multicast packets and one router detects the other routers’ duplicate packets. Use the following protocol names to set this option: ospf; kernel; igrp; rip; static; bgp; direct and ospfase. The values assigned to each protocol must match for each router on a multi-access LAN.

`assert-rank protocol` *`protocol name`* `rank default`	Specifies default assert-rank values for supported protocols that match other implementations. The direct default value is 0. The ospf default value is 10; the kernel default value is 40; the static route default value is 60; the IGRP default value is 80; the rip default value is 100; the bgp default value is 170.

Show PIM Commands

Use these commands to monitor and troubleshoot PIM. These commands apply to both dense-mode and sparse-mode implementations.

```
show pim
        interfaces
        interfaces if_address
        neighbors
        neighbor ip_address
        memory
        timers
        stats
        summary
```

The following show commands apply only to sparse-mode PIM implementations.

```
show pim
        bootstrap
        candidate-rp
        joins
        rps
        sparse-mode-stats
        group-rp-mapping <mcast_address>
```

The following show commands apply only to PIM when IP clustering is enabled.

```
show pim
        networks
        network ip_address
```

Route Aggregation

Use the following group of commands to take numerous specific routes and aggregate them into one encompassing route. Route aggregation potentially reduces the number of routes advertised by a given protocol.

Only the receiver uses aggregate routes to forward packets. A router that receives a packet that does not match one of the component routes that resulted in the generation of an aggregate route responds with an Internet Control Message Protocol (ICMP) network unreachable message. This message prevents packets or unknown component routes from following a default route to another network where they would be continually forwarded back to the border router until their TTL expires.

Create an aggregate route by first specifying the network address and mask length. Second, provide a set of contributing routes. To define a contributing route, specify a source (routing protocol, static route, or interface route) and a route filter (an IP prefix). An aggregate route can have many contributing routes, but at least one of the routes must be present to generate an aggregate.

```
set aggregate ip_prefix
        contributing protocol protocol contributing-route
           <all | ip_prefix> <on | off>
        contributing protocol protocol contributing-route <ip_prefix>
           exact on
        contributing protocol protocol contributing-route ip_prefix
           refines on
        off
        contributing protocol <protocol> off
        rank default
        rank <0-255>
        weight default
        aspath-truncate <on | off>
```

Arguments

`contributing protocol` *`protocol`* `contributing-route <all | ip_prefix <on | off>`	Specifies the IP address and mask length of the new aggregate route and the contributing protocol or interface route. To specify a protocol, enter *`direct`*, *`static`*, *`ospf2`*, *`ospf2ase`*, *`bgp`*, *`rip`*, *`igrp`*, *`rip`*, or *`aggregate`*. To specify a contributing route, enter *`all`* to contribute all the routes for a specific protocol or enter the IP address and mask length to contribute a specific route.
`contributing protocol` *`protocol`* `contributing-route` *`ip_prefix`* `exact on`	Specifies the IP address and the mask length of the new aggregate route and the contributing protocol and its corresponding IP address and mask length. The designation exact on means that the the contributing route is limted to the specifed IP address and mask length only.
`contributing protocol` *`protocol`* `contributing-route` *`ip_prefix`* `refines on`	Specifies the IP address and mask length of the new aggregate route and the contributing protocol and its corresponding IP address and mask length. The designation refines on means that the contributing route is based on addresses with a greater value than the specified mask length of the specified IP address. You cannot enable both exact on and refines on at the same time. If you enable refines on when exact on is enabled, exact on is automatically disabled.
`rank default`	Specifes the rank to assign to the aggregate route when routes from different protocols to the same destination are present. For each route, the route from the protocol with the lowest rank is used. Each routing protocol has a different default rank value. Aggregate routes have a default rank of 130.

`rank <0-255>`	Specifes the rank to assign to the aggregate route when routes from different protocols to the same destination are present. For each route, the route from the protocol with the lowest rank is used. Each routing protocol has a different default rank value.
`weight default`	Specifies a value that breaks a tie if select routes going to the same destination have the same rank value. The route with the highest weight is the active route. The active route is installed in the kernel forwarding table and redistributed to the other routing protocols. The default weight value is 0.
`weight <0-65535>`	Specifies a value that breaks a tie if select routes going to the same destination have the same rank value. The route with the highest weight is the active route. The active route is installed in the kernel forwarding table and redistributed to the other routing protocols.
`aspath-truncate <on | off>`	Specifies that the autonomous system (AS) path be truncated to the longest common AS path. The default, or off, option, Specifies building an AS path that consits of sets and sequences of all contributing AS paths. **Default:** off

BOOTP

Use this group of commands to set and view parameters for the bootstrap protocol.

BOOTP Interfaces

Use this group of commands to configure BOOTP properties for specific interfaces.

```
set bootp interface if_name
        primary ip_address wait-time <0-65535> on
        relay-to ip_address <on | off>
        off
```

Arguments

`primary ip_address wait-time <0-65535> on`	Specifies the ip_address to stamp as the gateway address on all BOOTP requests. The wait-time value Specifies the minimum amount of time, in seconds, to wait before forwarding a bootp request. Each client-generated bootp request includes the elapsed time since the client began the booting process. The bootp relay does not forward the request until the indicated elapsed time at least equals the specifed wait time. This delay provides an opportunity for a local configuration server to reply before attempting to relay to a remote server.
`relay-to ip_address <on | off>`	Specifies the server to which BOOTP requests are forwarded. You can specify more than one server.
`off`	Disables BOOTP on the specified interface.

BOOTP Show Commands

Use this group of commands to monitor and troubleshoot BOOTP implementation.

```
show bootp
        interfaces
        interface if_name
        stats
        stats receive
        stats request
        stats reply
```

DVMRP

Use the following group of commands to set and view parameters for DVMRP.

DVMRP Interfaces

Use the following commands to configure DVMRP properties for specific interfaces.

```
set dvmrp interface if_name
        <on | off>
        threshold <1-255>
        threshold default
        metric <1-32>
        metric default
```

Arguments

`<on | off>`	Specifies whether to run DVMRP on the specified interface
`threshold <1-255>`	Specifies the minimum time to live (TTL) required for a multicast packet to be forwarded. Note that the TTL of forwarded packets is only compared to the threshold; it is not decremented by the threshold. Every multicast router decrements the TTL by 1. The packet is forwarded only if the TTL of the packet is greater than the threshold set for the outbound port. When connecting to the Internet Multicast Backbone (MBONE) the following values are recommended. For a link within a set set the threshold value at 1; for a site boundary set the value at 32; for a regional boundary, set the value at 64; and for a continental boundary, set the value at 128.
`threshold default`	Specifies a value of 1.

`metric <1-32>`	Specifies the cost associated with sending a packet on the interface. It may be used to influence the choice of routes. A less expensive interface (smaller metric) is preferred to a more expensive interface (larger metric). You should use the smallest possible metric.
`metric default`	Specifies a value of 1.

DVMRP Timers

Use the following commands to configure values for DVMRP timers. Nokia recommends that if you have a core multicast network, configure the timer values so that they are uniform throughout a network. Otherwise, you can rely on the default timer values.

```
set dvmrp
        neighbor-probe-interval <5-30>
        neighbor-probe-interval default
        neighbor-timeout-interval <35-8000>
        neighbor-timeout-interval default
        route-report-interval <10-2000>
        route-expiration-time <20-4000>
        route-expiration-time default
        route-holddown-period <0-8000>
        route-holddown-period default
        cache-lifetime <60-86400>
        cache-lifetime default
```

Arguments

`neighbor-probe-interval <5-30>`	Specifies the interval, in seconds, at which probe messages are sent on each DVMRP interface. **Default**: 10
`neighbor-probe-interval default`	Specifies a value of 10 seconds

`neighbor-timeout-interval <35-8000>`	Specifies the interval, in seconds, after which a silent neighbor is timed out. For DVMRPv3 neighbors, the default is 35, and for non-DVMRPv3 neighbors, the default is 140.
`neighbor-timeout-interval default`	For DVMRPv3 neighbors, the default is 35, and for non-DVMRPv3 neighbors, the default is 140.
`route-report-interval <10-2000>`	Specifies the interval, in seconds, at which routing updates are sent on each DVMRP interface. **Default**: 60.
`route-report-interval default`	Specifies a value of 60 seconds.
`route-expiration-time <20-4000>`	Specifies the interval, in seconds, after which a route that has not been refreshed is placed in the route hold-down queue. **Default**: 140.
`route-expiration-time default`	Specifies a value of 140 seconds.
`route-holddown-period <0-8000>`	Specifies the interval, in seconds, for which routes in the hold-down queue are advertised with a metric of infinity before they are deleted. **Default**: 120.
`route-holddown-period default`	Specifies a value of 120 seconds.
`cache-lifetime <60-86400>`	Specifies the interval, in seconds, that a cached forwarding entry is maintained in the kernel forwarding table before it is timed out because of inactivity. **Default**: 300.
`cache-lifetime default`	Specifies a value of 300 seconds.

DVMRP Show Commands

Use the following commands to monitor and troubleshoot your DVMRP implementation.

```
show dvmrp
        interfaces
        interfaces if_name
        neighbors
        neighbor ip_address
        stats
        mfc
        reports
        route
        neighbor-routes
        summary
```

Static Routes

Static routes cause packets moving between a source and a destination to take a specified next hop. Static routes allow you to add routes to destinations that are not described by dynamic routing protocols. A static route can also be useful in providing a default route.

Configuring Static Routes

Use the following group of commands to configure specific static routes.

```
set static-route ip_prefix
        nexthop gateway address gateway_address priority <1-8> on
        nexthop gateway logical gateway_address priority <1-8> on
        nexthop gateway address gateway_address off
        nexthop gateway logical gateway_address off
        nexthop reject
        nexthop blackhole
        off
        rank default
        rank <0-255>
```

Arguments

`nexthop gateway address` *`gateway_address`* `priority <1-8> on`	Specifies the static route and the gateway address. The gateway address is an IP address or a logical interface. If your gateway address is a logical interface, enter the interface name. If the your gateway address is an unnumbered interface, use its logical interface as the gateway address. The priority value determines the order in which the next hops are selected and multiple next hops are defined with different priorities. Switching over to the next hop in the list happens only when an interface fails. Switching over does not happen for "non-reachability" next hops if the interface state is still up. If the route has the same priority as another, and the corresponding interface is up, the route is an equal-cost, multipath route. Lower priority next hops are preferred. You must configure a priority value. This option does not have a default value.

`nexthop logical` *`if_name`* `priority <1-8> on`	Specifies the static route and the logical gateway. For a logical gateway, enter the interface name. For example, if your gateway is an unnumbered interface, use its logical interface as the gateway. The priority value determines the order in which the next hops are selected and multiple next hops are defined with different priorities. Switching over to the next hop in the list happens only when an interface fails. Switching over does not happen for "non-reachability" next hops if the interface state is still up. If the route has the same priority as another, and the corresponding interface is up, the route is an equal-cost, multipath route. Lower priority next hops are preferred. You must configure a priority value. This option does not have a default value.
`nexthop gateway address` *`gateway_address`* `off`	Disables the gateway address only for the IP address configured as the endpoint of the static route from your system. This option does not delete the route itself.
`nexthop gateway logical` *`if_name`* `off`	Disables the gateway only for the logical interface configured as the endpoint of the static route from your system. This option does not delete the route itself.
`nexthop reject`	Specifies for packets to be dropped rather than forwarded and for unreachable messages to be sent to the packet originators. Specifying this option causes this route to be installed as a reject route.
`nexthop blackhole`	Specifies for packets to be dropped rather than forwarded. Unlike reject option, however, the blackhole option does not result in unreachable messages being sent to the packet originators.
`off`	deletes the specified static route and deletes any next hops associated with the route.

`rank default`	Specifies the rank for the specified static route the routing system uses to determine which route to use when there are routes from different protocols to the same destination. For each route, the route from the protocol with the lowest rank number is used. The default rank for static routes is 60.
`rank <0-255>`	Specifies the rank for the specified static route the routing system uses to determine which route to use when there are routes from different protocols to the same destination. For each route, the route from the protocol with the lowest rank number is used.

Use the following commands to define an existing default static route. To establish a new default route, use the commands in the preceding section to create a new static route and then use the set static-route default command to disable the old default static route.

```
set static-route default
        next hop gateway address gateway_address priority <1-8> on
        nexthop gateway logical gateway_address priority <1-8> on
        nexthop gateway address gateway_address off
        nexthop gateway logical gateway_address off
        nexthop reject
        nexthop blackhole
        ip_prefix off
        ip_prefix rank default
        ip_prefix rank <0-255>
```

Static Multicast Routes

When Protocol Independent Multicast (PIM) is enabled, PIM expects packets to arrive on the reverse-path forwarding (RPF) interface, that is, the interface used to reach the source of the multicast data. PIM also checks the RPF to learn which interface it should use to send join/prune messages. With builds of IPSO 4.2 previous to Build 078, the RPF interface is always identified by the unicast routing table.

Build 078 and later includes static multicast routes, which you can use to provide an alternative route table to use for the RPF check. If both a static multicast route and a unicast route are available for a specific destination, PIM uses the static multicast route.

Static multicast routes allow PIM to be independent of unicast routing and let you deploy topologies in which multicast and unicast traffic flow over different paths. For instance, if you want to balance your traffic load by separating the path used by HTTP traffic from the path used by streaming stock quotes, you could configure a static multicast route to the source network that specifies a next hop gateway address that is different from the next hop address (for the same source) in the unicast routing table.

Use the following commands to configure static multicast routes, use the following commands:

```
set static-mroute <sender_IP_address/mask | default>
        nexthop gateway
               address gateway_address <on | off | priority <1-8>>
               logical logical_interface priority <on | off | priority
                  <1-8>>
        off

show static-mroute
```

Arguments

sender_IP_address/mask	Specifies the address and network mask of the multicast sender.
default	Specifies a default gateway to use for RPF lookups.
gateway_address	Specifies the address of the gateway router.
logical_interface	Specifies the name of the logical interface that connects to the gateway router.

`priority <1-8>`	Specifies the order in which the next hops are selected when there are multiple next hops for the same destination. The next hop with the lowest priority value is used. If the interface for this next hop fails, IPSO uses the next hop with the next lowest value.
`on`	Enables the specified multicast static route.
`off`	Disables the specified multicast static route.

ICMP Router Discovery

Use this group of commands to set and view parameters for the ICMP router discovery protocol.

ICMP Router Discovery Interfaces

Use the following commands to configure router discovery properties for specific interfaces.

```
set rdisc interface if_name
        <on | off>
        min-adv-interval <3-1800>
        min-adv-interval default
        max-adv-interval <4-1800>
        max-adv-interval default
        adv-lifetime integer
        adv-lifetime default
        advertise ip_address <on | off>
        advertise ip_address preference ineligible
        advertise ip_address preference integer
```

Arguments

`<on | off>`	Specifies whether to run ICMP router discovery on a specified interface
`min-adv-interval <3-1800>`	Specifies the minimum time (in seconds) allowed between sending unsolicited broadcast or multicast ICMP router advertisementson the interface.
`min-adv-interval default`	Specifies a value of 450 seconds.
`max-adv-interval <4-1800>`	Specifies the maximum time (in seconds) allowed between sending unsolicited broadcast or multicast ICMP router advertisements on the interface.
`max-adv-interval default`	Specifies a value of 600 seconds
`adv-lifetime` *`integer`*	Specifies the time (in seconds) placed in the lifetime field of router advertisement packets sent from the interface. Enter an interger value between the configured value for the maximum advertisement interval and 9000.
`adv-lifetime default`	Specifies a value of 1800 or 3 times the configured maximum advertisement interval.
`advertise` *`ip_address`* `<on | off>`	Specifies whether to advertise the specified IP address that is associated with the interface should be advertised in router advertisement packets.
`advertise` *`ip_address`* `preference ineligible`	Specifies not to use the specified IP address as a default router.
`advertise` *`ip_address`* `preference` *`integer`*	Specifies the preferability of the specified IP address as a default router address relative to other router addresses on the same subnet.

ICMP Router Discovery Show Commands

Use the following commands to monitor and troubleshoot your ICMP router discovery implementation.

```
show rdisc
        interfaces
        interface if_name
        stats
        summary
```

IP Broadcast Helper

Use the following group of commands to set and view parameters for IP Broadcast Helper.

IP Broadcast Helper Forwarding

Use the following commands to control whether to forward packets that are not locally originated by a source directly on the receiving interface.

```
set iphelper
        forward-nonlocal <on | off>
```

IP Broadcast Helper Interfaces

Use the following commands configure IP Broadcast Helper properties for specific interfaces.

```
set iphelper interface if_name
        off
        udp-port <1-65535> off
        udp-port <1-65535> relay-to ip_address <on | off>
```

Arguments

`forward-nonlocal <on | off>`	Enter *on* to specify that packets be forwarded even if the source is not directly on the receiving interface. Enter *off* to require that packets for relay be generated by a source that is directly on the receiving interface. **Default:** off
`interface <if_name> off`	Specifies to disable the interface configured for iphelper
`udp-port <1-65535> off`	Specifies to disable the UDP services configured for this interface.
`udp-port <1-65535> relay-to ip_address <on | off>`	Specifies the UDP services defined for forwarding on the interface. Client UDP packets with the specified UDP port number are forwarded to the configured server(s). The IP address for the UDP port Specifies a new server to send client packets received for the associated interface and UDP service.

IP Broadcast Helper Show Commands

Use these commands to monitor and troubleshoot your IP Helper implementation.

```
show iphelper
        services
        stats
```

Network Time Protocol

Use the following commands to set and view parameters for network time protocol (NTP). NTP lets you synchronize time among different machines.

Configuring an NTP Server

```
set ntp
        server ip_address version <1-3>
        prefer server ip_address
        peer ip_address version <1-3>
        prefer peer ip_address
        master source ip_address stratum <0-15>
```

Adding an NTP Server

Use the following commands to add a new NTP server.

```
add ntp
        server ip_address version <1-3>
        prefer server ip_address
        peer ip_address version <1-3>
        prefer peer ip_address
```

Deleting an NTP Server

Use the following commands to delete an NTP server.

```
delete ntp
        server ip_address
        peer ip_address
```

Arguments

`server ip_address version <1-3>`	Specifies the IP address of the time server from which your system synchronizes its clock. The specified time server does **not** synchronize to the local clock of your system. The version number Specifies which version of NTP to run. Nokia recommends that you run version 3. **Default:** 3

`prefer server` *ip_address*	Specifies to prefer the specified time server if more than one configured time server is functioning.
`peer` *ip_address* `version <1-3>`	Specifies the IP address of the time server from which this system synchronizes its clock. The specified peer time time can syncrhonize to the local clock of your system. The version number Specifies which version of NTP to run. Nokia recommends that you run version 3. **Default:** 3
`prefer peer` *ip_address*	Specifies to prefer the specified peer time server if more than one configured time server is functioning
`master source` *ip_address* `stratum <0-15>`	Specifies to use this system as the source of time. Enter the system's <ip_address>. Stratum Specifies the number of hops away from a correct source of time this system's clock should appear to be. **Default:** 0 **NOTE:** Nokia recommends that you maintain this default value.

NTP Show Commands

Use the following commands to monitor and troubleshoot your NTP implementation.

```
show ntp
        active
        ntp master
        ntp peer ip_address
        ntp peers
        ntp server ip_address
        ntp servers
```

Dial on Demand Routing

Use the following commands to create, delete, or view the configuration of a dial on demand routing (DDR) list and add or delete ISDN interfaces to it. If you do not assign an ISDN interface to a DDR list, any traffic passed to the interface will cause it to attempt to set up a connection.

Dial on Demand Routing Commands

Use the commands in this section to configure DDR lists, and rules for the DDR lists.

```
add ddrlist name

show ddrlist

delete ddrlist name

add ddrlist name interface log_if_name

delete ddrlist name interface log_if_name
```

Arguments

name	Specifies a unique name for the DDR list. Names can contain letters, numbers, and underscores but must begin with a letter and must be no longer than 15 characters.

Use the following commands to create rules that you assign to DDR lists. These rules tell the system which packets should trigger it to set up or maintain an ISDN connection. When you create a DDR list, a default rule is automatically created for it.

```
add ddrlist name rule rule_num
        action <skip | ignore | accept>
        src-address ip_address
        src-masklen <0-32>
        dest-address ip_address
        dest-masklen <0-32>
        src-port <0—65535>
        dst-port <0—65535>
        protocol name
```

Use the following commands to configure DDR rules. Before you can configure DDR rules, you must assign a logical ISDN interface to the DDR list by entering `add ddrlist` *`name`* `interface` *`log_if_name`*.

```
set ddrlist name rule rule_num
        action <skip | ignore | accept>
        src-address ip_address
        src-masklen <0-32>
        dest-address ip_address
        dest-masklen <0-32>
        src-port <0—65535>
        dst-port <0—65535>
        protocol name

delete ddrlist name rule rule_num
```

The default values shown in the following table apply automatically created default rule.

Arguments

`add ddrlist` *`name`* `rule` *`rule_num`*	*`rule_num`* is the number of an existing rule. The new rule being created will be positioned and numbered before the existing rule (and the number of the existing rule will be incremented by one).The default rule's number is `1 until you create additional rules`.

`set ddrlist` *`name`* `rule` *`rule_num`*	When you use the `set` version of these commands, *`rule_num`* is the number of the rule you want to modify.
`delete ddrlist` *`name`* `rule` *`rule_num`*	Deletes the rule with the number of *`rule_num`*.
`action <skip | ignore | accept>`	If `action` is set to `accept`, the system will set up a connection when it encounters packets that match the rules' criteria. If `action` is set to `ignore`, matching packets will be passed over an existing connection but will not trigger the initiation of a connection. If `action` is set to `skip`, the system will not use compare packets against the rule—the rule is turned off. **>>** **Default:** skip
`src-address` *`ip_address`*	Specifies a source IP address to match against this rule.
`src-masklen <0-32>`	Specifies a mask length for the source IP address. **Default:** 0
`dest-address` *`ip_address`*	Specifies a destination IP address to match against the rule.
`dest-masklen <0-32>`	Specifies a mask length for the destination IP address. **Default:** 0
`src-port <0—65535>`	Specifies a specific port or range of ports for the source of a connection.This argument is only valid if the protocol for the rule is TCP, UDP, or `any`.

`dst-port <0—65535>`	Specifies a specific port or range of ports for the destination of a connection.This argument is only valid if the protocol for the rule is TCP, UDP, or `any`.
`protocol` *name*	Specifies the IP protocol that the rule applies to. Only one protocol can be specified per rule (unless you use the default value of `any, in` which case the rule applies to all protocols). This argument is not case-sensitive.

Routing Option Commands

Use the commands in this section to configure a variety of miscellaneous options that affect routing.

Equal-cost Path Splitting (Load Sharing)

Use the following command to specify a value for the maximum number of equal-cost paths that will be used when there is more than one equal-cost path to a destination. Only OSPF routes and Static routes are able to use more than one "next hop".

```
set max-path-splits <1—8>
```

Arguments

`max-path-splits <1—8>`	Indicates the maximum number of equal-cost paths that will be used when there is more than one equal-cost path to a destination. **Default:** 8

Use the following command to determine which “next hop” algorithm is used for forwarding when there is more than one “next hop” to a particular destination.

```
set nexthop-selection
        src-dest-hash
        dest-hash
        src-hash
        rr
```

Arguments

`src-dest-hash`	Source/destination hash: The IP forwarding code performs a hash function on the source and destination IP address of each packet that is forwarded to a multipath destination. This result is used to determine which next hop to use.
`dest-hash`	Destination hash: Operates the same as source/destination hash but only the destination IP address is used. Packets that are being sent to the same destination address will all use the same "next hop ".
`src-hash`	Source hash: Operates the same as source/destination hash but only the source IP address is used. Packets that are being sent from the same source address will all use the same "next hop".
`rr`	Round robin: Each time a set of "next hop"s is used for forwarding, a different "next hop" is used in a round-robin manner. This results in equal load sharing, but it is not recommended because it may result in out-of-order packet delivery for the same session.

Protocol Rank

Rank is used by the routing system when there are routes from different protocols to the same destination. For each route, the route from the protocol with lowest rank number will be used.

```
set protocol-rank protocol
        bgp rank <0–255>
        bgp rank default
        igrp rank <0–255>
        igrp rank default
        rip rank <0–255>
        rip rank default

set protocol-rank protocol
        ospf rank <0–255>
        ospf rank default
        ospfase rank <0–255>
        ospfase rank default
```

Arguments

protocol rank <0–255>	Specifies the protocol rank value.
bgp rank default	The default rank value for BGP is 170.
igrp rank default	The default rank value for IGRP is 80.
rip rank default	The default rank value for RIP is 100.
ospf rank default	The default rank value for OSPF is 10.
ospfase rank default	The default rank value for OSPF ASE routes is 150.

Trace Routing Commands

The routing system can optionally log information about errors and events. Logging is configured for each protocol or globally. Logging is not generally turned on during normal operations, as it can decrease performance. Log messages are saved in /var/log/ipsrd.log.

Configuring the Trace Log File

Use the following commands to configure the log file options for trace routing.

```
set tracefile
        size <1—4095>
        size default
        maxnum <1—4294967295>
        maxnum default
```

Arguments

`size <1—4095>`	Limits the maximum size of the trace file to the specified size, in megabytes.
`size default`	The default maximum trace file size is 1 MB.
`maxnum <1—4294967295>`	When the trace file reaches the specified size, it is renamed to file.0, then file.1, file.2, up to the maximum number of files.
`maxnum default`	The default maximum number of trace files is 10.

Trace Option Variables

You can specify a variety of different trace options with the `trace` command. While there are trace options specific to each protocol, many protocols share a set of options. These common trace options are specified in the *`traceoption`* variable. The following table lists the *`traceoption`* parameters.

Arguments

`all`	Trace all of the options in *`traceoptions`*.
`general`	Trace both `normal` and `route`.

`normal`	Trace normal protocol occurrences. Abnormal protocol occurrences are always traced.
`policy`	Trace the application of protocol- and user-specified policy to routes being imported and exported.
`route`	Trace routing table changes for routes installed by this protocol or peer.
`state`	Trace state machine transitions in the protocols.
`task`	Trace system interface and processing associated with this protocol or peer.
`timer`	Trace timer usage by this protocol or peer.

Use the following command to turn BGP trace options on or off.

```
set trace bgp
        keepalive <on | off>
        open <on | off>
        update <on | off>
        packets <on | off>
        traceoptions <on | off>
```

Arguments

`keepalive`	Trace BGP keepalive messages
`open`	Trace BGP open packets. These packets are sent between peers when they are establishing a connection.
`update`	Trace update packets. These packets provide routing updates to BGP systems.
`packets`	Trace all BGP protocol packets.
`traceoptions`	<all | general | normal | policy | route | state | task | timer>

Use the following command to turn DVMRP trace options on or off.

```
set trace dvmrp
        graft <on | off>
        mfc <on | off>
        mapper <on | off>
        neighbor <on | off>
        probe <on | off>
        prune <on | off>
        report <on | off>
        packets <on | off>
        traceoptions <on | off>
```

Arguments

`graft`	Trace DVMRP graft and graftack packets.
`mfc`	Trace DVMRP multicast forwarding cache packets
`mapper`	Trace DVMRP neighbor and neighbor2 packets.
`neighbor`	Trace DVMRP neighbor packets.
`probe`	Trace DVMRP probe packets.
`prune`	Trace DVMRP prune packets.
`report`	Trace DVMRP route report packets.
`packet`	Trace all DVMRP packets.
`traceoptions`	<all | general | normal | policy | route | state | task | timer>

Use the following command to turn ICMP trace options on or off.

```
set trace icmp
        error <on | off>
        info <on | off>
        routerdiscovery <on | off>
        packets <on | off>
        traceoptions <on | off>
```

Arguments

`error`	Trace only ICMP error packets, which include: • time exceeded • parameter problem • unreachable • source quench
`info`	Trace only ICMP informational packets, which include: • mask request/response • info request/response • echo request/response • time stamp request/response
`routerdiscovery`	Trace only ICMP router discovery packets.
`packets`	Trace all ICMP packets.
`traceoptions`	<all | general | normal | policy | route | state | task | timer>

Use the following command to turn IGRP trace options on or off.

```
set trace igrp
        packets <on | off>
        traceoptions <on | off>
```

Arguments

`packets`	Trace all IGRP packets.

traceoptions	<all | general | normal | policy | route | state | task | timer>

Use the following command to turn IGMP trace options on or off.

```
set trace igmp
        group <on | off>
        leave <on | off>
        mtrace <on | off>
        query <on | off>
        report <on | off>
        request <on | off>
        packets <on | off>
        traceoptions <on | off>
```

Arguments

group	Trace multicast group add, delete, refresh and accelerated leave.
leave	Trace IGMP “leave group” messages.
mtrace	Trace details of IGMP multicast traceroute request processing.
query	Trace IGMP membership query packets (both general and group-specific).
report	Trace IGMP membership report packets (both IGMPv1 and IGMPv2).
request	Trace IGMP multicast traceroute request packets.
packets	Trace all IGMP packets.
traceoptions	<all | general | normal | policy | route | state | task | timer>

Use the following command to turn IP broadcast helper trace options on or off.

```
set trace iphelper
        packets <on | off>
        traceoptions <on | off>
```

Arguments

`packets`	Trace all IP broadcast helper packets.
`traceoptions`	<all | general | normal | policy | route | state | task | timer>

Use the following command to turn MFC trace options on or off.

```
set trace mfc
        alerts <on | off>
        cache <on | off>
        interface <on | off>
        mcastdist <on | off>
        packets <on | off>
        resolve <on | off>
        wrongif <on | off>
        traceoptions <on | off>
```

Arguments

`alerts`	Trace multicast protocol alert callback functions.
`cache`	Trace log details of cache maintenance. These include: • addition or deletion of orphan entries (in other words, entries with no route to source). • addition or deletion of normal entries. • cache state aging and refresh.

`interface`	Trace log changes requested by external ipsrd modules (IGMP and multicast routing protocols) affecting the forwarding dependencies on an interface. These include: • addition or deletion of a forwarding interface due to routing changes. • changing of the parent (reverse path forwarding) interface due to routing changes.
`mcastdist`	Trace kernel multicast distribution entries. Both generic and PIM register encapsulation and decapsulation types.
`packets`	Trace all MFC related packets.
`resolve`	Trace kernel external resolve requests (both normal and PIM register types).
`wrongif`	Trace kernel multicast incoming interface violation notifications (both physical interface and PIM register types).
`traceoptions`	<all | general | normal | policy | route | state | task | timer>

Use the following command to turn PIM trace options on or off.

```
set trace pim
        assert <on | off>
        bootstrap <on | off>
        crp <on | off>
        graft <on | off>
        hello <on | off>
        join <on | off>
        mfc <on | off>
        mrt <on | off>
        packets <on | off>
        rp <on | off>
        register <on | off>
        trap <on | off>
        traceoptions <on | off>
```

Arguments

`assert`	Trace PIM assert messages.
`bootstrap`	Trace bootstrap messages (sparse-mode only).
`crp`	Trace candidate-RP-advertisements (sparse-mode only).
`graft`	Trace graft and graft acknowledgment packets.
`hello`	Trace PIM router hello packets.
`join`	Trace PIM join/prune messages.
`mfc`	Trace interaction with multicast forwarding cache.
`mrt`	Trace PIM multicast routing table events.
`packets`	Trace all PIM packets.
`rp`	Trace RP-specific events. This includes both RP set-specific and bootstrap-specific events (sparse-mode only).
`register`	Trace register and register-stop packets (sparse-mode only).
`trap`	Trace PIM trap messages.
`traceoptions`	<all | general | normal | policy | route | state | task | timer>

Use the following command to turn BGP trace options on or off.

```
set trace rip
        packets <on | off>
        traceoptions <on | off>
```

Arguments

`packets`	Trace all RIP packets.
`traceoptions`	<all | general | normal | policy | route | state | task | timer>

Use the following command to turn VRRP trace options on or off.

```
set trace vrrp
        advertise <on | off>
        traceoptions <on | off>
```

Arguments

advertise	Trace all VRRP packets.
traceoptions	<all | general | normal | policy | route | state | task | timer>

Use the following command to turn ICMP router discovery trace options on or off.

```
set trace router-discovery option <on | off>
        traceoptions
```

Arguments

traceoptions	<all | general | normal | policy | route | state | task | timer>

Use the following command to turn global trace options on or off.

```
set trace global
        adv <on | off>
        parse <on | off>
        traceoptions <on | off>
```

Arguments

adv	Trace the allocation of and freeing of policy blocks.
parse	Trace the lexical analyzer and parser.
traceoptions	<all | general | normal | policy | route | state | task | timer>

Use the following command to turn kernel trace options on or off.

```
set trace kernel
        iflist <on | off>
        interface <on | off>
        packets <on | off>
        remnants <on | off>
        request <on | off>
        routes <on | off>
        traceoptions <on | off>
```

Arguments

`iflist`	Trace `iflist`, the interface list scan.
`interface`	Trace interface status messages that are received from the kernel.
`packets`	Trace packets that are read from the kernel
`remnants`	Trace remnants, which specify routes read from the kernel when the routing daemon starts.
`request`	Trace requests, which specify to add, delete, or change routes in the kernel forwarding table.
`routes`	Trace routes that are exchanged with the kernel, including add, delete, or change messages and add, delete, or change messages received from other processes.
`traceoptions`	<all | general | normal | policy | route | state | task | timer>

Use the following command to turn OSPF trace options on or off.

```
set trace ospf
        ack <on | off>
        dd <on | off>
        dr <on | off>
        hello <on | off>
        lsa <on | off>
        packets <on | off>
        request <on | off>
        spf <on | off>
        trap <on | off>
        update <on | off>
        traceoptions <on | off>
```

Arguments

`ack`	Trace link-state acknowledgment packets.
`dd`	Trace all database description packets.
`dr`	Trace designated router packets.
`hello`	Trace hello packets.
`lsa`	Trace link-state announcement packets.
`packets`	Trace OSPF packets.
`request`	Trace link-state request packets.
`spf`	Trace shortest-path-first (SPF) calculations.
`trap`	Traces OSPF trap packets.
`update`	Trace link-state updates packets.
`traceoptions`	<all | general | normal | policy | route | state | task | timer>

Show Route Summary Commands

Use the commands in this section to view summary information about routes on your system.

Route Summary Commands

Use the following command to show information about active, inactive or all (both active and inactive) routes on your system for BGP, IGRP and RIP protocols.

```
show route
        igrp
        rip
        bgp <aspath | communities | detailed | metrics | suppressed>
        inactive <bgp | igrp | rip>
        all <bgp | igrp | rip>
```

Use the following command to show information about active, inactive, or all routes on your system for the OSPF protocol.

```
show route
        ospf
        inactive ospf
        all ospf
```

Use the following command to show information about active, inactive and all aggregate routes on your system.

```
show route
        aggregate
        inactive aggregate
        all aggregate
```

Use the following command to show additional information about routes on your system.

```
show route
        all
        all direct
        all static
        direct
        inactive
        inactive direct
        inactive static
        static
        summary
        destination ip_address
        exact ip_prefix
        less-specific ip_prefix
        more-specific ip_prefix
```

Show Routing Daemon (IPSRD) Commands

Use the following commands to view general information recorded by the IPSO routing daemon (IPSRD).

```
show ipsrd
        memory
        resources
        krt
        version
```

Arguments

`memory`	Displays the memory usage of the routing daemon. It shows the information for each routing protocol running on the system. • Total memory usage • MFC- memory used for the multicast forwarding cache (MFC) • Core - memory used by IPSRD for its internal purpose • *`Protocol`* - memory used by the given protocol

`resources`	Displays the following system information: • Total uptime • Total user time • Total system time • Page faults • Page reclaims • File system writes • File system reads • Message writes • Message reads • Signals received • Total swaps • Voluntary context switches • Involuntary context switches
`krt`	Displays statistical information about the messages sent and received on the raw sockets between the kernel and IPSRD. • KRT interface message count • KRT interface message length • KRT route message count (rx) • KRT route message length (rx) • KRT route message count (tx) • KRT route message length (tx) • KRT route adds • KRT route changes • KRT route deletes
`version`	Displays the following system information: • IPSRD version • System start time • Current time • System uptime

Show MFC Commands

Use the following commands to view information about multicast forwarding cache (MFC) on your system.

```
show mfc
        cache
        summary
        interface
        orphans
        stats
```

Arguments

`cache`	Displays MFC state information.
`summary`	Displays the following MFC state information: • Number of interfaces enabled • Number of cache entries • Kernel forwarding entry limit • Number of kernel forwarding entries • Cache entry average lifetime • Prune average lifetime • Cache age cycle • Data rate update interval • Multicast protocol (instance)
`interface`	Displays MFC interface state information.
`orphans`	Displays MFC orphan state information.
`stats`	Displays various information about the following MFC properties: • Resolve task summary • Resolve requests • RPF failure notifications • MFC maintenance

10 Traffic Management Commands

This chapter describes the commands you use to configure traffic management functionality on your system and to view current settings.

Access Control List Commands

Access control lists (ACLs) sort incoming network traffic into discrete packet streams based on fields in the packet header. An access list contains a set of rules called the ruleset. When a packet matches a rule, the system executes the action specified in the rule. Using the access list CLI commands, you can configure an access list to control the traffic from one or more interfaces. Also, each access list can be associated with incoming or outgoing traffic from each interface.

ACL Node Commands

Use the following command to show all created ACLs in the system.

```
show acl
```

Use the following command to create a new, uniquely-named ACL. You have the option to create an association between the ACL rule and the specified logical interface. The interface binding is related to the traffic flow direction. You can specify a logical interface for outgoing traffic, incoming traffic, or both. If you do not specify a version or interface bindings, the ACL will automatically be for IPv4 traffic and have no interface bindings.

```
add acl
        name
        name version <ip | ip6>
        name outinterface if_name
        name ininterface if_name
```

Use the following command to create an association between an existing ACL rule and the specified logical interface. The interface binding is related to the flow direction. You can specify a logical interface for outgoing traffic, incoming traffic, or both.

```
set acl name
        outinterface if_name
        ininterface if_name
```

Use the following command to remove the ACL from live configuration or to delete the association between an ACL rule and an interface. Specifying only the ACL name deletes the whole ACL and all interface associations, if they exist. To delete the interface association, specify a logical interface for outgoing traffic, incoming traffic, or both.

```
delete acl
        name
        name outinterface if_name
        name ininterface if_name
```

Arguments

`acl` *name*	Specifies the name of the ACL. Use alphanumeric characters.
`version <ip4 | ip6>`	Specifies the protocol version, either `ip` (IPv4) or `ip6` (IPv6). This parameter is optional. **Default:** ip4

`outinterface` *`if_name`*	Specifies the output interface of the access list. Protocol support is checked against the interface. IPv4 is accepted on all interfaces unless the interface has the “IPv6 only” flag on.
`ininterface` *`if_name`*	Specifies the input interface of the access list. Protocol support is checked against the interface. IPv4 is accepted on all interfaces unless the interface has the “IPv6 only” flag on.

Use the following command to activate or deactivate the ACL bypass mode. Turning the bypass on allows you to bypass all traffic control blocks (for example, the classifier, meter, and policer).

```
set acl name bypass <on | off>
```

Arguments

`acl` *`name`*	Specifies the name of the ACL. Use alphanumeric characters.
`bypass <on | off>`	Set the traffic control block bypass: on or off.

ACL Ruleset Commands

Use the following command to show information about all existing ACL rulesets.

```
show aclrules
```

> **Note**
> Every ACL has a default rule that is originally in position 1. When you use the `show` command, the default rule will be marked as such. You cannot

delete the default rule. This rule can only accept “accept” and “drop” as its action values.

Use the following command to show information about a specific ACL ruleset.

```
show aclrule name
```

Arguments

`aclrule` *name*	Specifies the name of the ACL for which to display ruleset information. Use alphanumeric characters.

Use the following command to add an ACL ruleset for an existing ACL.

```
add aclrule name position integer
```

Arguments

`aclrule` *name*	Spcifies the name of the ACL where the new ruleset will be added. The ACL must already exist. Use alphanumeric characters.
`position` *integer*	Specifies the ruleset position number within the ACL. Position number specifies ruleset priority within the ACL.The highest ruleset priority is position_num = 1.

Use the following command to set ACL ruleset for the specified ACL. The default ruleset, which is marked as “default” when you use the `show` command, can only accept or drop as the action value.

```
set aclrule name position integer action
   <accept | drop | reject | prioritize | skip | shape> srcaddr
   ip_address/netmask destaddr ip_address/netmask srcport <0—65535>
```

```
destport <0—65535> protocol <any | tcp | udp | 0—255> tcp-estab
<yes | no> tos <0x0—0xff> dsfield <none | 0x00—0xff> qspec <none | 0—
7>
```

Arguments:

`aclrule` *`name`*	Specifies the name of the ACL. Use alphanumeric characters. The command checks the existence of the ACL.
`position` *`integer`*	Specifies the ruleset position number within the ACL. The command checks the existence of the rule.

`action <accept | drop | reject | prioritize | skip | shape>`	Specifies the action to take when the interface associate with the rule encounters a packet matching the rule. The default ruleset, which is marked as "default" when you use the `show` command, can only accept accept or drop as the action value. The aggregation class must be configured when rule has priority action. Actions are as follows: • accept - Best effort queuing for packet. • drop - No service for packet: the packet is dropped • reject - No service for packet, similar to drop, but ICMP error packet is sent to the source. • prioritize - See "Queue Class Commands" on page 495 • skip - Skip the specified rule. It does not apply. • shape - Rate shaping. See "Aggregation Class Commands" on page 494. **Default:** skip

`srcaddr` *`ip_address/netmask`*	Specifies the source IP address and netmask to be used for matching this rule. IPv4 • Range: dotted-quad 0-255.0-255.0-255.0-255/0-32 • Example: 192.168.50.1/24 • Default: 0.0.0.0/0 IPv6 • Range: IPv6 prefix format/0-126 • Example: 2222::1:2:3:4/0 • Default: ::/0 **Note:** This should be an IPV6 prefix format, not an IPv6 address. Therefore, the prefix/mask 222::223/65 is not valid, while 222::223/0 and 222::/65 are valid.
`destaddr` *`ip_address/netmask`*	Specifies the destination IP address and netmask to be used for matching this rule. IPv4 • Range: dotted-quad 0-255.0-255.0-255.0-255/0-32 • Example: 192.168.50.1/24 • Default: 0.0.0.0/0 IPv6 • Range: IPv6 prefix format/0-126 • Example: 2222::1:2:3:4/0 • Default: ::/0 **Note:** This should be an IPV6 prefix format, not an IPv6 address. Therefore, the prefix/mask 222::223/65 is not valid, while 222::223/0 and 222::/65 are valid.

`srcport <0–65535>`	Specifies the source port number or port range. The defalut is the entire range, 0–65535.
`destport <0–65535>`	Specifies the destination port number or port range.The defalut is the entire range, 0–65535.
`protocol <any | tcp | udp | 0–255>`	Specifies the IP protocol to be used for matching this rule. **Default:** any
`tcp-estab <yes | no>`	Specifies whether TCP is established. You may specify the TCP-Estab only when protocol is “any” or “6” or “TCP.” **Default:** no
`tos <any | 0x0–0xff>`	Specifies the type of service to be used for matching this rule. **Default:** any
`dsfield <none>`	Specifies the DiffServ codepoint (DSCP) with which to mark traffic that matches this rule. **Default:** none

`qspec <none | 0–7>`	Specifies a queue specifier to be used by the output scheduler for traffic matching this rule. **Note:** This field is inactive when the rule action is set to anything other than prioritize. **Note:** When the DSfield is set to one of the predefined codepoints (internetwork control, expedited forwarding, or best effort), then the QueueSpec field is not used and the parameter value is "none." **Default:** none

Use the following `set` command to associate an ACL rule with the specified aggregation class. Use the `delete` command to disassociate an ACL rule with the specified aggregation class. For information on aggregation class commands, see "Aggregation Class Commands" on page 494.

```
set aclrule name position integer aggrclass name
```

```
delete aclrule name position integer aggrclass name
```

Use the following command to delete an ACL ruleset from the specified ACL.

```
delete aclrule name position integer
```

Arguments

`aclrule` *`name`*	Spcifies the name of the ACL. Use alphanumeric characters.
`position` *`integer`*	Specifies the ruleset position number within the ACL.
`aggrclass` *`name`*	Specifies the name of the aggregation class. Use alphanumeric characters.

Aggregation Class Commands

An aggregation class provides the mechanism to meter traffic flows and shape or police them to a configurable rate. Use the commands in this section to create new or delete existing aggregation classes and to modify the mean rate or burstsize.

Set, Change, and View Aggregation Classes

Use the following command to show all existing aggregation classes.

```
show aggrclasses
```

Use the following command to show a specific aggregation class.

```
show aggrclass name
```

Arguments

`aggrclass name`	Specifies the name of the aggregation class to display. Use alphanumeric characters.

Use the following command to add an aggregation class together with its meanrate and burstsize.

```
add aggrclass name meanrate <10—10000000> burstsize <1500 —150000>
```

Use the following command to set a new meanrate, burstsize or both meanrate and burstsize values for an existing aggregation class.

```
set aggrclass name
        meanrate <10—10000000>
        burstsize <1500—150000>
```

Arguments

`aggrclass` *name*	Specifies the name of the aggregation class. Use alphanumeric characters.
`meanrate <10—10000000>`	Specifies the packet stream mean rate in kilobytes. Range 10-10000000 Kbps
`burstsize <1500—150000>`	Specifies the burst value of the packet stream in bytes. Range 1500 - 150000 bytes

Use the following command to delete an existing aggregation class.

```
delete aggrclass name
```

Arguments

`aggrclass` *name*	Specifies the name of the aggregation class to delete. Use alphanumeric characters.

Queue Class Commands

Queue classes are used as templates for queue structures which can be associated with physical interfaces. You may configure items such as the depth of the queues, assign logical names to some of the queues, and set up a queue specifier.

Set, Change, and View Queue Classes

Use the following command to show all existing queue classes.

```
show qclasses
```

Use the following command to show a specific queue class.

```
show qclass name
```

Arguments

`qclass name`	Specifies the name of the queue class to display. Use alphanumeric characters.

Use the following command to add a queue class.

```
add qclass name
```

Arguments

`qclass name`	Specifies the name of the queue class to add. Use alphanumeric characters.

Use the following command to specify the scheduling algorithm for a queue class. The specified queue class must exist before you use this command.

```
set qclass name type <strict | wrr | cas>
```

Use the following commands to set values for one or multiple queue class properties. The specified queue class must exist before you use these commands. You can set the queue specifier and queue length in the same command, but you must use a separate command to set the logical name for the queue priority.

```
set qclass name priority <0—7>
        name name
        qspec <0—5>
        qlength <0—256>
        weight  <0—8>
        dropper <tail | wred>
        maxth value
        minth value
        const <1—16>
        dr1 value
        dr2 value
        dr3 value
```

Arguments

`qclass` *name*	Specifies the name of the queue class. Use alphanumeric characters.
`type <strict | wrr | cas>`	Specifies the scheduling algorithm: • `strict`: strict priority • `wrr`: weighted round robin • `cas`: cascade **Default:** strict
`priority <0—7>`	Specifies the strict priority of queue. A lower priority value has greater preference for service. Three priorities are reserved for Internetwork Control, Expedited Forwarding and Best Effort traffic, which are priorities 0, 1, and 7, respectively

`name` *name*	Specifies the logical name for a priority level. It is used to help identify the use of a queue. • Format: Alphanumeric characters and underscore with no spaces • Default: The default values for priority queues 2 to 6 are: Priority 2 - Q_priority_2 Priority 3 - Q_priority_3 Priority 4 - Q_priority_4 Priority 5 - Q_priority_5 Priority 6 - Q_priority_6
`qspec <0—5>`	Queue specifier is used within the class as a logical identifier. Use the Queue Specifier with the classifier to direct traffic to a specific queue. The following specifiers are predefined: • 7 - Internetwork Control • 6 - Expedited Forwarding • 0 - Best Effort Identifiers 7, 6 and 0 are reserved for internetwork control, expedited forwarding and best effort respectively (RFC 791). • Range 0-5 **Default:** For queues 3 to 7 the default is 0

`qlength <0—256>`	Specifies the maximum number of packets that may be queued before packets are dropped (range 10 - 256). A value of zero (0) is used to disable a queue. Neither the Internetwork Control nor the Best Effort queue can be disabled. The range is zero (0) through 256. **Default:** Varies based on use of queue for queues 3 to 7. For NetControl the default is 16, for Expedited Forwarding the default is 32, and for Best Effort the default is 64.
`weight <0—8>`	Specifies the weight for a queue, which configures the proportions in which the link capacity is to be divided among the queues. If you use WRR, you can assign weight values of 0-8. For cascade scheduling, queue 7 and queue 6 must be assigned strict priority (weight 0) and the weights for the remaining queues must be in descending order. You can configure adjacent queues to have identical weights, but the weight assigned to a given queue cannot be greater than that assigned to a queue with a greater queue specifier. This option is not available if the scheduling algorithm is strict priority.
`dropper <tail | wred>`	Specifies the drop method: • tail: Drop any packets that arrive for a queue after the queue is full. • wred: Drop packets as configured by WRED options. This option is not available if the scheduling algorithm is strict priority.

`maxth` *value*	Maximum threshold: If the average queue length is greater than or equal to this value, all arriving packets are dropped. Applies only to WRED. **Default:** 64
`minth` *value*	Minimum threshold: If the average queue length is less than or equal to this value, no packets are dropped. Applies only to WRED. **Default:** 32
`const <1—16>`	Specifies how adaptive WRED is to traffic bursts. Choosing a higher value creates a slower moving average, which has the benefit of reducing the variation in queue length (and therefore reducing packet drops caused by traffic bursts) but causes the system to react slower to congestion. Choosing a lower value allows the system to react faster to congestion but might result in packet drops caused by overreactions to temporary bursts in traffic. Applies only to WRED. **Default:** 10
`dr1` *value*	Drop rate for low precedence traffic. If the value is 1024, IPSO drops one packet after servicing approximately 1024 packets. Applies only to WRED. **Default:** 1024
`dr2` *value*	Drop rate for medium precedence traffic. If the value is 512, IPSO drops one packet after servicing approximately 512 packets. Applies only to WRED. **Default:** 512

`dr3 value`	Drop rate for low precedence traffic. If the value is 256, IPSO drops one packet after servicing approximately 256 packets. Applies only to WRED. **Default:** 256

Use the following command to associate a queue class with a given physical interface.

```
set qclass name interface if_name qmode
   <disabled | maxthroughput | minlatency>
```

Arguments

`qclass name`	Specifies the name of the queue class. Use alphanumeric characters.
`interface if_name`	Specifies the name of the physical interface.
`qmode <disabled | maxthroughput | minlatency>`	Specifies the QoS queue mode for the interface. The options are: disabled, maximum throughput or minimal latency

Use the following command to delete a queue class.

```
delete qclass name
```

Arguments

`qclass name`	Specifies the name of the aggregation class to delete. Use alphanumeric characters.

Use the following command to view queue class statistics for interfaces associated with the queue class.

```
show qclass-statistics
```

ATM QoS

Asynchronous transfer mode (ATM) quality of service (QoS) descriptor configuration describes the traffic parameters for ATM virtual channels (VCs). The QoS configuration for an ATM VC is done by associating the VC with an ATM QoS descriptor.

ATM QoS descriptors belong to one of two categories: unspecified bit rate (UBR) or constant bit rate (CBR). The UBR does not have any QoS guarantees and is the default category used for an ATM VC in the absence of any explicit QoS descriptor association. The CBR does have some QoS guarantee. The CBR limits the maximum cell output rate to adhere to the requirements on CBR traffic that the network imposes. Use the commands in this section to:

- Add, delete, or show ATM QoS descriptors
- Add, delete, or show association of ATM QoS descriptors with ATM VCs
- Show available or reserved bandwidth on an ATM interface

Configuring ATM QoS Descriptors

Use the following command to add an ATM QoS descriptor with a specified peak cell rate:

```
add atmqos qosd name pcr <64—146000>
```

Use the following command to delete the specified ATM QoS descriptor:

```
delete atmqos qosd name
```

Use the following command to show all ATM QoS descriptors:

```
show atmqos qosd
```

Arguments

`qosd` *`name`*	Specifies the name of the ATM QoS descriptor. The category for any descriptor you create is CBR.
`pcr <64–146000>`	Specifies the maximum cell rate, in kilobits per second, used in the output direction on a CBR channel. Peak cell rate is rounded down to a multiple of 64 kbps. One cell per second corresponds to 424 bits per second.

Use the following command to associate an ATM QoS descriptor with an ATM VC on the specified physical interface:

```
set atmqos interface if_name vc integer qosd name
```

Use the following command to delete the association of an ATM QoS descriptor with a VC on the specified physical ATM interface:

```
delete atmqos interface if_name vc <vpc/vci | vci>
```

Arguments

`interface` *`if_name`*	Specifies the name of the ATM physical interface on which the VC is riding. Example: atm-s3p1
`vc <`*`vpc/vci	vci`*`>`	Specifies the VC to associate or disassociate with the ATM QoS descriptor. Use non-negative integers in a VPI/VCI format (for example, 1/3) or a single integer to represent the VCI. The VPI value defaults to zero if not specified.
`qosd` *`name`*	Specifies the name of the ATM QoS descriptor to associate with the VC.

Use the following commands to show all ATM QoS descriptor and VC associations on the specified physical ATM interface:

```
show atmqos interface if_name settings
```

Use the following command to show available or reserved bandwidth on the specified physical ATM interface:

```
show atmqos interface if_name bandwidth
   <available | reserved>
```

DSCP to VLAN Priority Commands

Differentiated Services Code Point (DSCP) to virtual LAN (VLAN) priority mapping allows you to utilize fixed class of service (CoS) values on your network. You can map the DSCP of the IP and IPv6 packets to VLAN priority tags in the egress (outgoing) direction. In the ingress (incoming) direction, no mapping occurs from VLAN priority tags to DSCPs.

You can enable and disable mapping by using CLI commands, but you cannot configure the DSCP to VLAN priority values. The DSCP values correspond to the following CoS values:

- DSCP 0–7 = CoS 0
- DSCP 8–15 = CoS 1
- DSCP 16–23 = CoS 2
- DSCP 24–31 = CoS 3
- DSCP 32–39 = CoS 4
- DSCP 40–47 = CoS 5
- DSCP 48–55 = CoS 6
- DSCP 56–63 = CoS 7

When you enable mapping, it is done similarly for each Ethernet frame that has a VLAN tag, regardless of the VLAN ID.

Configuring DSCP to VLAN Mapping

Use the following command to enable or disable mapping of DSCP to VLAN priority service to this system.

```
set custom dscp-to-vlanprio <on | off>
```

Use the following command to show the status of mapping between VLAN priority and DSCP service on this system.

```
show custom dscp-to-vlanprio
```

11 Monitoring Commands

This chapter describes the system monitoring commands that you can enter from the CLI prompt.

If you use Tab command completion for certain monitoring commands, you see `relative` listed as a possible option. Do not use this option. It is used internally when an IPSO cluster is present to allow Cluster Voyager to display aggregated data for the cluster.

Current and Historical Network Reports

Use the commands in the following sections to configure various system reports.

Saving Reports to Files

You can save system reports to files and specify a delimiter to separate fields in the reports. For example, to create a memory utilization report and download it to a file, you could use the following command:

```
Show monitor summary hourly memoryutilization delimiter <, | ; | Tab>
   filename name
```

Report files are saved in the directory /var/log/monitor.

Configuring How Much Data is Stored

Use the following commands to configure and see how much historical data is collected on the system:

```
set monitor config maxhour <24-167>

show monitor config maxhour
```

Arguments

`maxhour <24-167>`	Specifies how many hours worth of collected data is stored on the system. Data that is older than the specified number of hours is deleted. This option controls how much data is available when you use `starttime/endtime` form of the `show monitor` commands. It does not affect how much data is available when you use the `summary` form of the `show monitor` commands. Nokia recommends that you set this option to 24 hours on diskless systems to avoid exhausting the available storage space. **Default:** 24

Configuring CPU Utilization Reports

Use the following commands to turn data collection on or off and to set the data collection time interval.

```
set monitor config
        cpuutilization state <on | off>
        cpuutilization interval <60-2100000>
```

Use the following commands to view whether data collection is on or off and the data collection time interval.

```
show monitor config
        cpuutilization state
        cpuutilization interval
```

Use the following command to specify a start time and end time for the CPU utilization report.

```
show monitor
        starttime <date time year> endtime <date time year>
           cpuutilization
        summary <hourly | daily | weekly | monthly> cpuutilization
```

Configuring Memory Utilization Reports

Use the following commands to turn data collection on or off and to set the data collection time interval.

```
set monitor config
        maxhour
        memoryutilization state <on | off>
        memoryutilization interval <60-2100000>
```

Use the following commands to view whether data collection is on or off and the data collection time interval.

```
show monitor config
        maxhour
        memoryutilization state
        memoryutilization interval
```

Use the following command to specify a start time and an end time for the memory utilization report.

```
show monitor
        start time <date time year> endtime <date time year>
           memoryutilization
        summary <hourly | daily | weekly | monthly> memoryutilization
```

Arguments

`state <on | off>`	Turns data collection on or off. **Default:** on
`interval <60-2100000>`	Specifies the data collection time in seconds. **Default:** 60
`starttime <date time year>`	Specifies the start time for a data collection report. You must enter the date, time and year between quotes. The following is an example of how you enter the date: “Oct 27 02:21:55 2001”

Configuring Interface Linkstate Reports

Use the following commands to turn date collection on or off and to set the data collection time interval.

```
set monitor config
        linkstate state <on | off>
        linkstate interval <60-2100000 seconds>
```

Use the following commands to view whether data collection is on or off and the data collection time interval.

```
show monitor config
        linkstate state
        linkstate interval
```

Use the following command to specify a linkstate report start time and end time for a given interface.

```
show monitor
        starttime <date time year> endtime <date time year> linkstate
           interface-type <logical | physical>
           interface <name>
```

Use the following command to view the state of a given interface over a specified period of time.

```
show monitor summary <hourly | daily | weekly | monthly> linkstate
   interface-type <logical | physical>
   interface <name>
```

Configuring Rate Shaping Bandwidth Reports

Use the following commands to turn data collection on or off and to set the data collection time interval.

```
set monitor config
        rateshape type <bytesdelayed | packetdelayed> state <on | off>
        rateshape interval <60-2100000 seconds>
```

Use the following commands to view whether data collection is on or off and the data collection time interval.

```
show monitor config
        rateshape type <bytesdelayed | packetdelayed> state
        rateshape interval
```

Use the following command to specify a rate shaping bandwidth report start time and end time for a given rate shape data and aggregation class.

```
show monitor starttime <date time year> endtime <date time year>
   rateshape type <bytesdelayed | packetdelayed>
   aggregate <name>
```

Use the following command to view rate shaping data over a specified period of time.

```
show monitor summary <hourly | daily | weekly | monthly> rateshape
   type <bytesdelayed | packetdelayed>
   aggregate <name>
```

Configuring Interface Throughput Reports

Use the following commands to turn date collection on or off and to set the data collection time interval.

```
set monitor config
        throughput type <bytes | packets | multicast | broadcast>
           state <on | off>
        throughput interval <60-2100000 seconds>
```

Use the following commands to view whether data collection is on or off and the data collection time interval.

```
show monitor config
        throughput type <bytes | packets | multicast | broadcast>
           state
        throughput interval
```

Use the following command to specify a interface throughput report start time and end time for a given interface.

```
show monitor starttime <date time year> endtime <date time year>
   throughput type <bytes | packets | multicast | broadcast>
   interface-type <logical | physical> interface <name> network
   <ip_address>
```

Use the following command to view the state of a given interface over a specified period.

```
show monitor summary <hourly | daily | weekly | monthly> throughput
   type <bytes | packets | multicast | broadcast> interface-type
   <logical | physical> interface <name>
```

Arguments

`aggregate <name>`	Specifies information for a given aggregation class. The name is case-sensitive.
`endtime <date time year>`	Specifies the start time for a data collection report. You must enter the date, time and year between quotes. The following is an example of how you enter the date: “Oct 27 02:21:55 2001”
`interface <name>`	Specifies the name of a given interface of which you want to collect data.
`interface-type <logical | physical>`	Specifies whether you want to collect data from the logical or physical interface.
`interval <60-2100000>`	Specifies the data collection time in seconds. **Default:** 60
`rateshape type <bytesdelayed | packetdelayed>`	Specifies the type of rate shaping data.
`state <on | off>`	Turns data collection on or off. **Default:** on
`starttime <date time year>`	Specifies the start time for a data collection report. You must enter the date, time and year between quotes. The following is an example of how you enter the date: “Oct 27 02:21:55 2001”
`throughput type <bytes | packets | multicast | broadcast>`	Specifies the type of data that you want to collect.

Useful System Information

Use the following commands to view various information about your system.

Displaying Useful System Statistics

Use the following commands to display useful system statistics.

```
show useful-stats
```

The useful system statistics summarize configuration information such as the following:

- The number of configured active routes
- The number of forwarded packets
- The number of configured VRRP masters
- The percentage of real memory in use
- The percentage of system disk space used on the system

Displaying Interface Settings

Use the following command to display interface settings.

```
show interfacemonitor
```

Arguments

`interfacemonitor`	Displays the interface settings.

Displaying System Logs

Use the following commands to display system logs.

```
show
        logging
        logininfo all
        logininfo user
        log auditlog
        log httpd-access-log
        log httpd-error-log
        log messagelog
               type name date name keyword name case-sensitive
               include-zipped name
```

Arguments

`logging`	Displays the system log configuration.
`logininfo all`	Displays the login/logout activity for all users.
`logininfo` *`name`*	Dislplays the login/logout activity for the named user.
`log auditlog`	Displays a log that shows configuration changes made by users.
`log httpd-access-log`	Displays the httpd_access_log generated by the web-server, which shows access to the platfomr made through the web server. The information includes the client IP address, time of access, and page accessed.
`log httpd-error-log`	Displays the log file generated by the web server showing errors that the web server detects.
`log messagelog`	Displays the complete system-wide log messages or a search can be performed based on type of log messages, log dates a particular keyword etc. Searches can case-sensitive or case-insensitive. Archived log files can be included in the search.

`type` *`name`* `date` *`name`* `keyword` *`name`* `case-sensitive` `include-zipped` *`name`*	Specifies the type of log messages to be displayed. The type is the severity of the log message, from least sever to most sever, and can be specified by one of the following keywords: • emerg • alert • crit • error • warning • notice • info • debug To include all kinds of log message either specify the keyword "all" or skip this part of the command. If date is specified only messages matching that date are displayed. The date is specified in one of the following forms: Oct OR "Oct 12" that is, short form of month or a combination of month and date. Date cannot be specified without specifying a month. Enter a keyword to search for in the system-wide log file. By default the search is case-insensitive. To do a case-sensitive search, specify the word "case-sensitive" in the command. Enter a list of comma-separated zipped files to be included in the search, or just specify the keyword "all" to include all available zipped files. If a list is specified it should be of the form: messages.1.gz,messages.0.gz,messages.2.gz No spaces are allowed in the list.

Displaying Interface Traffic Statistics

Use the following commands to display interface traffic statistics.

```
show iftrafficstats
```

The interface traffic statistics contains information about the state of the device represented by the physical and logical interface.

Physical interface information includes the following:

- Physical, which is the name of the physical interface
- Up, which shows the running state of the physical interface
- InBytes, which is the input bytes counter for the physical interface
- OutBytes, which is the output bytes counter for the physical interface
- InErrs, which is the input error counter for the physical interface
- OutErrs, which is the output error counter for the physical interface

Logical interface information includes the following:

- Logical, which is the name of the logical interface
- Active, which is the logical interface administrative status
- Up, which is the running state of the logical interface
- Type, which is the type of device or virtual circuit accessed through the logical interface (for example, Ethernet, ATM, FDDI).
- InBytes, which is the input bytes counter for the logical interface.
- OutBytes, which is the output bytes counter for the logical interface.

Displaying the Interface Monitor

Use the following commands to display the interface monitor.

```
show interfacemonitor
```

The interface monitor provides information such as the following:

- Interface name
- Status (up or down)
- Logical name
- State (multiple conditions where present)
- Maximum transmission Unit (MTU)
- Up to down transitions

Displaying Resource Statistics

Use the following commands to display resource statistics.

```
show resource-statistics
```

Resource statistics include the following:

- Total uptime
- Total user time
- Total system time
- Major page faults
- Minor page faults
- File system writes
- File system reads
- Message writes
- Message reads
- Signals received
- Total swaps
- Voluntary context switches
- Involuntary context switches

Displaying the Forwarding Table

Use the following commands to display the forwarding table.

```
show forwarding-table
```

Note
The `show forwarding-table` command displays only the default instance forwarding table.

Forwarding table information includes the following:

- Destination, which is the destination host or network
- Gateway, which is the IP address or the interface name of the outgoing interface that is the next-hop device through which packets should be routed
- Flags, which is information about the route stored as binary choices. The mapping between codes and flags is as follows:

Code	Flag
1	RTF_PROTO1: Protocol specific routing flag 1
2	RTF_PROTO2: Protocol specific routing flag 2
3	RTF_PROTO3: Protocol specific routing flag 3
B	RTF_BLACKHOLE: Just discard packets during updates
C	RTF_CLONING: Generate new routes on use
c	RTF_PRCLONING: Protocol-specified generate new routes on use
D	RTF_DYNAMIC: Created dynamically by redirect
G	RTF_GATEWAY: Destination requires forwarding by intermediary
H	RTF_HOST: Host entry net otherwise
L	RTF_LLINFO: Valid protocol to link address translation
M	RTF_MODIFIED: Modified dynamically by redirect
R	RTF_REJECT: Host or net unreachable
S	RTF_STATIC: Manually added
U	RTF_UP: Route usable
W	RTF_WASCLONED: Route was generated as a result of cloning
X	RTF_XRESOLVE: External daemon translates proto to link address

- Netif, which is the name of the local interface

Displaying Hardware Monitors

The commands described in this section display the status for various system components. Components for which the status can be displayed include temperature sensors, watchdog timers, voltage sensors, power supplies, and fan sensors. The command returns status only for installed components.

Use the following commands to display all system status information.

```
show sysenv all
```

Displaying a Temperature Sensor Information

Use the following commands to display temperature sensor information.

```
show sysenv temperature
        all
        sensor sensor-number
           <all | location | status | current | limit | hysteresis>
```

Temperature sensor information includes the following:

- Location
- Status
- Current value
- Temp limit
- Temp hysteresis

Arguments

`all`	Displays information about all of the temperature sensors.
`sensor` *`sensor-number`* `<all | location | status | current | limit | hysteresis>`	Displays information about the temperature sensor specified by *`sensor-number`*. `all` displays all information about the sensor specified by *`sensor-number`*. `location` displays the location (for example, lm79_temp) of the sensor specified by *`sensor-number`*. `status` displays the status (for example, good) for the sensor specified by *`sensor-number`*. `current` displays the current value of the sensor specified by *`sensor-number`*. `limit` displays the limit value for the sensor specified by *`sensor-number`*. `hysteresis` displays the hysteresis value for the sensor specified by *`sensor-number`*.

Displaying a Watchdog Timer Information

Use the following commands to display watchdog timer information.

```
show sysenv watchdog-timer
   <all | status | mode | tickles | last-reboot>
```

Watchdog timers track a specific amount of time. If that time runs out before an expected event occurs, some action, such as a reboot or an error-message generation, is triggered.

Watchdog timer information includes the following:

- Present
- Status
- Mode
- Tickles
- Last reboot

Arguments

`all`	Displays all watchdog timer information.
`status`	Displays the watchdog timer status.
`mode`	Displays the watchdog timer mode.
`tickles`	Displays watchdog timer tickle information.
`last-reboot`	Displays the watchdog timer last-reboot time.

Displaying Voltage Sensor Information

Use the following commands to display voltage sensor information.

```
show sysenv voltage
        all
        sensor sensor-number
           <all | location | status | nominal | measured | error | lo-
           limit | hi-limit>
```

Voltage sensor information includes the following:

- Location
- Status
- Nominal
- Measured
- Error
- Low Limit
- High Limit

Arguments

`all`	Displays all voltage sensor information.
`sensor` *`sensor-number`* `<all | location | status | nominal | measured | error | lo-limit | hi-limit>`	Displays information about the voltage sensor specified by *`sensor-number`*. `all` displays all information about the sensor specified by *`sensor-number`*. `location` displays the location (for example, lm79_IN0) of the sensor specified by *`sensor-number`*. `status` displays the status of the sensor specified by *`sensor-number`* `nominal` displays the nominal value for the sensor specified by *`sensor-number`*. `measured` displays the measurement for the sensor specified by *`sensor-number`*. `error` displays error information as a numerical value for the sensor specified by *`sensor-number`*. `lo-limit` displays the low-limit for the sensor specified by *`sensor-number`*. `hi-limit` displays the high-limit for the sensor specified by *`sensor-number`*.

Displaying Power Supply Information

Use the following commands to display power supply information.

```
show sysenv power-supply
        all
        id pwr-supply-id
           <all | present | volts | amps | status | revision>
```

Power supply information includes the following:

- Present (yes or no)
- Volts
- Amps
- Status (for example, OK)
- Revision

Hyphens are displayed where no information is available for an information field.

Arguments

`all`	Displays all voltage sensor information
`id` *`pwr-supply-id`* `<all | present | volts | amps | status | revision>`	`all` displays all information about the power supply specified by *`pwr-supply-id`*. `present` indicates whether or not the power supply specified by *`pwr-supply-id`* is present. `volts` displays voltage information for the power supply specified by *`pwr-supply-id`*. `amps` displays amperage information for the power supply specified by *`pwr-supply-id`*. `status` displays status information (for example, OK) for the power supply specified by *`pwr-supply-id`*. `revision` displays the revision for the power supply specified by *`pwr-supply-id`*.

Displaying Fan Sensor Information

Use the following commands to display fan sensor information.

```
show sysenv fan
        all
        sensor sensor-number
           <all | location | status | current | normal | limit>
```

Fan sensor information includes the following:

- Location (for example, lm79_FAN1)
- Status (for example, Normal)
- Current Value
- Normal Value

Arguments

`all`	Displays all fan sensor information
`sensor` *`sensor-number`* `<all | location | status | current | normal | limit>`	`all` displays all information about the fan sensor specified by *`sensor-number`*. `location` displays the location (for example, lm79_FAN1) of the fan sensor specified by *`sensor-number`*. `status` displays status information (for example, Normal) for the fan sensor specified by *`sensor-number`*. `current` displays current information for the fan sensor specified by *`sensor-number`*. `normal` displays normal information for the fan sensor specified by *`sensor-number`*. `limit` displays limit information for the fan sensor specified by *`sensor-number`*.

Displaying a Network Interface Card Slot Status

Use the following command to display the network interface card (NIC) slot status for each slot in the system.

```
show sysenv slot-status
```

Network interface card slot status information includes the following:

- State (for example, empty or online)
- Driver
- ID (for example, DEC_21152)

12 Command-Line Utilities

This chapter contains information on a selected list of command-line utilities. On the whole, these command-line utility descriptions are similar to the man pages you would access with the man command on a typical Unix-based system.

apcssd

Name

`apcssd` - the command interface to the APC Simple Signaling Daemon, SSD.

Synopsis

```
/etc/apcssd start
/etc/apcssd stop
/etc/apcssd restart
```

Description

The APC Simple Signaling Daemon, SSD, provides basic shutdown and notification services for systems powered by an APC UPS in simple signaling mode. When an extended power failure occurs, SSD sends notifications to logged-in users and shuts down the appliance cleanly.

The options for apcssd are as follows:

`start` Starts the SSD daemon.

`stop` Stops the SSD daemon.

`restart` Stops the running daemon and then starts a new one.

Configuration File

The configuration file for SSD is /etc/ssd.conf, which contains two parameters:

`TTY` The serial port used to connect the simple signaling cable to the UPS. The default is /dev/ttyd1.

`MaxBatRT` The number of seconds the system should run after power fails. The default is 180 seconds (3 minutes). To allow the system to run as long as possible before the battery discharges, set this to a large number.

Notes

- To start the SSD daemon automatically on reboot, enter the apcssd start command into /var/etc/rc.local.
- Make sure you use the 940-0020B simple signaling cable to connect the UPS to your appliance. Do not use a standard RS-232 cable or a smart signaling cable.
- A Smart UPS, such as SU420NET, must be in simple mode. If it is in smart mode:

  1 Reattach the 940-0024C smart signaling cable and use a terminal program to communicate with the UPS.

  2 Make sure the settings are 2400 baud, 8 bits, no parity, and no hardware or software flow control.

  3 Type the Y character, which should return "SM". Now type a capital R (case matters). The UPS should respond with "BYE." It is now in simple signaling mode.

  4 Reattach the 940-0020B cable.

cst

Name

`cst` - configuration summary tool

Synopsis

```
cst [-small]
```

Description

The Nokia IPSO Configuration Summary Tool (CST) is a troubleshooting tool that displays a summary of the current configuration of your system. It also creates a compressed file (in `/admin`) that you can move to a workstation and uncompress to produce various HTML and image files that make it easy to view the current configuration.

CST replaces `ipsoinfo`, and its HTML output is easier to read than a typical `ipsoinfo` output file.

CST reports the MD5 of key files on the system. This is important if vmcores exist on the system and an analysis is to be performed. The MD5 information reported by CST ensures the correct kernel and loadable modules are used during the analysis.

CST generates historical reports for cpu utilization, memory utilization, and traffic throughput for all interfaces. These reports contain pie and chart graphs as well as the raw data collected. CST also gathers debug information on VRRP, RIP, OSPF, BGP, and DVMRP.

The following option is available:

`-small` Do not gather core files. If you do not specify this option, CST gathers core files produced in the last four weeks.

df

Name

`df` - display free disk space

Synopsis

```
df [-ikn] [-t fstype] [file | filesystem ...]
```

Description

`df` displays statistics about the amount of free disk space on the specified filesystem or on the filesystem of which file is a part. Values are displayed in 512-byte per block counts. If neither a file or a filesystem operand is specified, statistics for all mounted filesystems are displayed (subject to the -t option below).

The following options are available:

`-i` Include statistics on the number of free inodes.

`-k` Use 1024-byte (1-Kbyte) blocks rather than the default. Note that this overrides the BLOCKSIZE specification from the environment.

`-n` Print out the previously obtained statistics from the filesystems. This option should be used if it is possible that one or more filesystems are in a state such that they will not be able to provide statistics without a long delay. When this option is specified, df will not request new statistics from the filesystems, but will respond with the possibly stale statistics that were previously obtained.

`-t` Only print out statistics for filesystems of the specified types. More than one type may be specified in a comma separated list. The list of filesystem types can be prefixed with ‘‘no’’ to specify the filesystem types for which action should not be taken. For example, the df command: df -t nonfs,mfs lists all filesystems except those of type NFS and MFS.

Environment

Blocksize

If the environment variable BLOCKSIZE is set, the block counts will be displayed in units of that size block.

Bugs

The `-n` and `-t` flags are ignored if a file or filesystem is specified.

See Also

lsvfs(1), quota(1), fstatfs(2), getfsstat(2), statfs(2), getmntinfo(3), fstab(5), mount(8), quot(8)

History

A df command appeared in Version 1 AT&T UNIX.

ipsoinfo

This utility is replaced by the CST utility. Executing the command `ipsoinfo` invokes CST.

netstat

Name

`netstat` - show network status

Synopsis

```
netstat [-Aan] [-f address_family] [-M core] [-N system]
netstat [-bdghimnrs] [-f address_family] [-M core] [-N system]
netstat [-bdn] [-I interface] [-M core] [-N system] [-w wait]
netstat [-p protocol] [-M core] [-N system] netstat [-F]
```

Description

The netstat command symbolically displays the contents of various network-related data structures. There are a number of output formats, depending on the options for the information presented. The first form of the command displays a list of active sockets for each protocol. The second form presents the contents of one of the other network data structures according to the option selected. Using the third form, with a wait interval specified, netstat will continuously display the information regarding packet traffic on the configured network interfaces. The fourth form displays statistics about the named protocol.

The options have the following meaning:

`-A`	With the default display, show the address of any protocol control blocks associated with sockets; used for debugging.
`-a`	With the default display, show the state of all sockets; normally sockets used by server processes are not shown.

`-b`	With either interface display (option -i or an interval, as described below), show the number of bytes in and out.
`-d`	With either interface display (option -i or an interval, as described below), show the number of dropped packets.
`-f` *`address_family`*	Limit statistics or address control block reports to those of the specified address family. The following address families are recognized: inet, for AF_INET, ns, for AF_NS, iso, for AF_ISO, and unix, for AF_UNIX.
`-F`	Show the state of the flows table.
`-g`	Show information related to multicast (group address) routing. By default, show the IP Multicast virtual-interface and routing tables. If the -s option is also present, show multicast routing statistics.
`-h`	Show the state of the IMP host table (obsolete).
`-I` *`interface`*	Show information about the specified interface; used with a wait interval as described below.
`-i`	Show the state of interfaces which have been auto-configured (interfaces statically configured into a system, but not located at boot time are not shown). If the -a options is also present, multicast addresses currently in use are shown for each Ethernet interface and for each IP interface address. Multicast addresses are shown on separate lines following the interface address with which they are associated.
`-M`	Extract values associated with the name list from the specified core instead of the default /dev/kmem.
`-m`	Show statistics recorded by the memory management routines (the network manages a private pool of memory buffers).

`-N`	Extract the name list from the specified system instead of the default /kernel.
`-n`	Show network addresses as numbers (normally netstat interprets addresses and attempts to display them symbolically).This option may be used with any of the display formats.
`-p`	protocol Show statistics about protocol, which is either a well-known name for a protocol or an alias for it. Some protocol names and aliases are listed in the file /etc/protocols. A null response typically means that there are no interesting numbers to report. The program will complain if protocol is unknown or if there is no statistics routine for it.
`-s`	Show per-protocol statistics. If this option is repeated, counters with a value of zero are suppressed.
`-r`	Show the routing tables. When -s is also present, show routing statistics instead.
`-w` *`wait`*	Show network interface statistics at intervals of wait seconds.

The default display, for active sockets, shows the local and remote addresses, send and receive queue sizes (in bytes), protocol, and the internal state of the protocol. Address formats are of the form “host.port” or “network.port” if a socket’s address specifies a network but no specific host address. When known the host and network addresses are displayed symbolically according to the data bases /etc/hosts and /etc/networks, respectively. If a symbolic name for an address is unknown, or if the -n option is specified, the address is printed numerically, according to the address family. For more information regarding the Internet “dot format,” refer to inet(3)). Unspecified, or “wildcard”, addresses and ports appear as “*”.

The interface display provides a table of cumulative statistics regarding packets transferred, errors, and collisions. The network addresses of the interface and the maximum transmission unit (“mtu”) are also displayed.

The routing table display indicates the available routes and their status. Each route consists of a destination host or network and a gateway to use in forwarding packets. The flags field shows a collection of information about the route stored as binary choices. The individual flags are discussed in more detail in the route(8) and route(4) manual pages. The mapping between letters and flags is:

1	RTF_PROTO2	Protocol specific routing flag #1
2	RTF_PROTO1	Protocol specific routing flag #2
3	RTF_PROTO3	Protocol specific routing flag #3
B	RTF_BLACKHOLE	Just discard pkts (during updates)
C	RTF_CLONING	Generate new routes on use
c	RTF_PRCLONING	Protocol-specified generate new routes on use
D	RTF_DYNAMIC	Created dynamically (by redirect)
G	RTF_GATEWAY	Destination requires forwarding by intermediary
H	RTF_HOST	Host entry (net otherwise)
L	RTF_LLINFO	Valid protocol to link address translation
M	RTF_MODIFIED	Modified dynamically (by redirect)
R	RTF_REJECT	Host or net unreachable
S	RTF_STATIC	Manually added
U	RTF_UP	Route usable
W	RTF_WASCLONED	Route was generated as a result of cloning
X	RTF_XRESOLVE	External daemon translates proto to link address

Direct routes are created for each interface attached to the local host; the gateway field for such entries shows the address of the outgoing interface. The refcnt field gives the current number of active uses of the route. Connection oriented protocols normally hold on to a single route for the duration of a connection while connectionless protocols obtain a route while sending to the same destination. The use field provides a count of the number of packets sent using that route. The interface entry indicates the network interface utilized for the route.

When netstat is invoked with the -w option and a wait interval argument, it displays a

running count of statistics related to network interfaces. An obsolescent version of this option used a numeric parameter with no option, and is currently supported for backward compatibility. This display consists of a column for the primary interface (the first interface found during autoconfiguration) and a column summarizing information for all interfaces. The primary interface may be replaced with another interface with the -I option. The first line of each screen of information contains a summary since the system was last rebooted. Subsequent lines of output show values accumulated over the preceding interval.

See Also

iostat(1), nfsstat(1), ps(1), vmstat(1), hosts(5), networks(5), protocols(5), services(5), trpt(8), trsp(8)

History

The netstat command appeared in 4.2BSD.

Bugs

The notion of errors is ill-defined.

ping

Name

`ping` - send ICMP ECHO_REQUEST packets to network hosts

Synopsis

```
ping [ -CFILNORUdfjmnqruv ] [ -c count ] [ -K count ] [ -g | -G
gateway ] [ -t timeout ] [ -l preload ] [ -o type ] [ -p pattern
] [ -s length ] [ -T ttl ] [ -S tos ] host
default_multicast_interface
```

Description

The DARPA Internet is a large and complex aggregation of network hardware, connected together by gateways. Tracking a single-point hardware or software failure can often be difficult. Ping utilizes the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams ("pings") have an IP and ICMP header, followed by a struct timeval, and then an arbitrary number of "pad" bytes used to fill out the packet. Default datagram length is 64 bytes, but this may be changed using the command-line option.

The default mode is to send a packet every timeout seconds and display the response, including sequence number and round-trip time if the packet size allows.Two other modes are available and are mutually exclusive:

`-C`	Use cisco style packet sending; that is, spew packets as fast as possible, but don’t send packet N+1 until an reply or timeout has been registered for packet N. A ‘!’ is printed for every response received and a ‘.’ is printed when a packet is not received during the timeout period. Also, a ‘*’ is printed whenever a duplicate response is received.
`-F`	Floodping style packet sending; send packets as fast as possible, attempting to send at at least 100 packets per second. A ‘.’ is printed for every response that is missed and a ‘*’ is printed whenever a duplicate packet is received. Only users in group zero (0) are allowd to use this option.
`-G`	Specify strict source routing via this gateway. Multiple gateways may be specified by repeating the option. This option only works on BSD 4.3 based systems that allow setting of IP options.
`-I`	On systems where setting the TTL of outbound ICMP packets is supported, this option sets the initial TTL to 1 and increments it each time an ECHO_REQUEST packet is sent, until an ECHO_REPLY is received or the TTL wraps to zero.
`-K`	Count Terminate after receiving responses for count packets.This is subtly different from -c. If you use this option to ping a host that is down, the command might not terminate.
`-L`	On systems supporting IP multicasting, disable multicast loopback.
`-N`	Always lookup the returned IP addresses. By default the IP source address of packets received is printed in numeric form. Use of this option can adversely affect the round trip statistics.
`-O`	Print Options. The contents of the IP Options on packets sent and received are listed.

`-R`	Insert “record route” IP option in outgoing packets, summarizing routes taken when program exits. This option only works on BSD 4.3 based systems that allow setting of IP options.
`-S tos`	On systems where setting the TOS of outbound ICMP packets is supported, this option sets the TOS field of all outbound ICMP ECHO_REQUEST packets.
`-T ttl`	On systems where setting th TTL of outbound ICMP packets is supported, this option sets the initial TTL of all outbound ICMP ECHO_REQUEST packets.
`-U`	Print round trip times and accumulate statistics with millisecond precision.
`-c count`	Terminate after count packets have been sent or received.
`-d`	Enable socket level debugging with the SO_DEBUG option.
`-f` *`Fast`*	ping, send a packet as soon as a response is received.
`-g`	Specify loose source routing via this gateway. Multiple gateways may be specified by repeating the option. This option only works on BSD 4.3 based systems that allow setting of IP options.
`-j`	When the destination host is really an IP multicast group on systems supporting IP multicasting, join the specified group.
`-l`	Specifies the number of packets to preload; packets which are sent at startup before listening for a response. The deault is not to send any preload packets.
`-m`	Enable printing of a summary of missed responses.
`-n`	Disable hostname lookup of the returned IP addresses. By default the hostname and IP address of the destination host is displayed in the header, summry and “record route” summary. If the hostname lookup fails, just the IP address is displayed.

`-o` Specifies the type of ICMP packet to send. The default is an ECHO_REQUEST. The type may be abreviated down to one character.

Supported types are:

`echo`	Send an ECHO_REQUEST, expect and ECHO_RESPONSE. If at least eight bytes of data are available, round trip timing is done. This is the default.
`info`	Send an INFO_REQUEST, expect and INFO_RESPONSE.Roundtrip timing and additional data are not possible with this type.
`mask`	Send a MASK_REQUEST, expect a MASK_REPLY. Round trip timing and additional data are not possible with this type.
`router`	Send a ROUTER_SOLICITATION, expect a ROUTER_ADVERTISEMENT. Round trip timing and additional data are not possible with this type.
`timestamp`	Send a TIMESTAMP_REQUEST, expect a TIMESTAMP_REPLY. Route trip and additional timing is always done. Additional data may be carried.

`-p` Specifies the pattern to fill unused data in the packets.The default is to fill with the data position modulo 256. A pattern Is a string of hex digit pairs used to specify the contents of the bytes of the packet. Multiple patterns are concatenated. The pattern(s) are replicated until they fill the packet.

`-q` Disable displaying of ping response messages, only display the summary.

`-r`	Bypass the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it (e.g., after the interface was dropped by routed(8C)).
`-t`	Specify the timeout between pings in normal and cisco style modes as a floating point number. The default is one second. Only users in group zero (0) are allowed to specify a value less than one second.
`-u`	Print round trip times and accumulate statistics with microsecond precision.
`-v`	Verbose output. ICMP packets other than ECHO_RESPONSE that are received are listed.

When using ping for fault isolation, it should first be run on the local host, to verify that the local network interface is up and running. Then, hosts and gateways further and further away should be “pinged.” Ping sends one datagram per second, and prints one line of output for every ECHO_RESPONSE returned. No output is produced if there is no response. If an optional length is specified, it is used as the length of the data portion of the ICMP ECHO_REQUEST packet. The default length is 56 data bytes. If an optional count is given, only that number of requests is sent. Round-trip times and packet loss statistics are computed. When all responses have been received or the program times out (with a count specified), or if the program is terminated with a SIGINT, a brief summary is displayed.

This program is intended for use in network testing, measurement and management. It should be used primarily for manual fault isolation. Because of the load it could impose on the network, it is unwise to use ping during normal operations or from automated scripts.

Return Values

An exit status of zero is returned if at least one response was heard from the specified host; a status of two if the transmission was successful but no responses were received; or another value if an error occurred.

Author

Mike Muuss

Co-conspirators

Ron Natalie, David Paul Zimmerman, Jeffrey C Honig, Vernon Schryver, Dennis Ferguson.

Bugs

More options than ls(1).

See Also

netstat(1), ifconfig(8C)

ps

Name

`ps` - process status

Synopsis

```
ps [-aCehjlmrSTuvwx] [-M core] [-N system] [-O fmt] [-o fmt]
[-p pid] [-t tty] [-W swap] ps [-L]
```

Description

Ps displays a header line followed by lines containing information about your processes that have controlling terminals. This information is sorted by controlling terminal, then by process ID.

The information displayed is selected based on a set of keywords (see the -L -O and -o options). The default output format includes, for each process, the process' ID, controlling terminal, CPU time (including both user and system time), state, and associated command.

The process file system (see procfs(5)) should be mounted when ps is executed, otherwise not all information will be available.

The options are as follows:

`-a`	Display information about other users' processes as well as your own.
`-C`	Change the way the cpu percentage is calculated by using a ‘‘raw’’ cpu calculation that ignores ‘‘resident’’ time (this normally has no effect).

`-e`	Display the environment as well.
`-h`	Repeat the information header as often as necessary to guarantee one header per page of information.
`-j`	Print information associated with the following keywords: user, pid, ppid, pgid, sess, jobc, state, tt, time and command.
`-L`	List the set of available keywords.
`-l`	Display information associated with the following keywords: uid, pid, ppid, cpu, pri, nice, vsz, rss, wchan, state, tt, time and command.
`-M`	Extract values associated with the name list from the specified core instead of the default /dev/kmem.
`-m`	Sort by memory usage, instead of by process ID.
`-N`	Extract the name list from the specified system instead of the default /kernel.
`-O`	Add the information associated with the space or comma separated list of keywords specified, after the process ID, in the default information display. Keywords may be appended with an equals (‘‘=’’) sign and a string. This causes the printed header to use the specified string instead of the standard header.
`-o`	Display information associated with the space or comma separated list of keywords specified. Keywords may be appended with an equals (‘‘=’’) sign and a string. This causes the printed header to use the specified string instead of the standard header.
`-p`	Display information associated with the specified process ID.
`-r`	Sort by current cpu usage, instead of by process ID.
`-S`	Change the way the process time is calculated by summing all excited children to their parent process.

`-T`	Display information about processes attached to the device associated with the standard input.
`-t`	Display information about processes attached to the specified terminal device.
`-u`	Display information associated with the following keywords: user, pid, %cpu, %mem, vsz, rss, tt, state, start, time and command. The -u option implies the -r option.
`-v`	Display information associated with the following keywords: pid, state, time, sl, re, pagein, vsz, rss, lim, tsiz, %cpu, %mem and command. The -v option implies the -m option.
`-W`	Extract swap information from the specified file instead of the default /dev/drum.
`-w`	Use 132 columns to display information, instead of the default which is your window size. If the -w option is specified more than once, ps will use as many columns as necessary without regard for your window size.
`-x`	Display information about processes without controlling terminals.

A complete list of the available keywords are listed below. Some of these keywords are further specified as follows:

`%cpu`	The cpu utilization of the process; this is a decaying average over up to a minute of previous (real) time. Since the time base over which this is computed varies (since processes may be very young) it is possible for the sum of all %CPU fields to exceed 100%.
`%mem`	The percentage of real memory used by this process.
`flags`	The flags associated with the process as in the include file <sys/proc.h>:

P_ADVLOCk	0x00001	Process may hold a POSIX advisory lock

P_CONTROLT	0x00002	Has a controlling terminal
P_INMEM	0x00004	Loaded into memory
P_NOCLDSTOP	0x00008	No SIGCHLD when children stop
P_PPWAIT	0x00010	Parent is waiting for child to exec/exit
P_PROFIL	0x00020	Has started profiling
P_SELECT	0x00040	Selecting; wakeup/waiting danger
P_SINTR	0x00080	Sleep is interruptible
P_SUGID	0x00100	Had set id privileges since last exec
P_SYSTEM	0x00200	System proc: no sigs, stats or swapping
P_TIMEOUT	0x00400	Timing out during sleep
P_TRACED	0x00800	Debugged process being traced
P_WAITED	0x01000	Debugging process has waited for child
P_WEXIT	0x02000	Working on exiting
P_EXEC	0x04000	Process called exec
P_NOSWAP	0x08000	Another flag to prevent swap out
P_PHYSIO	0x10000	Doing physical I/O
P_OWEUPC	0x20000	Owe process an addupc() call at next ast
P_SWAPPING	0x40000	Process is being swapped

`lim` The soft limit on memory used, specified via a call to setrlimit(2).

`lstart` The exact time the command started, using the ‘‘%c’’ format described in strftime(3).

`nice` The process scheduling increment (see setpriority(2)).

`rss` The real memory (resident set) size of the process (in 1024 byte units).

`start` The time the command started. If the command started less than 24 hours ago, the start time is displayed using the ‘‘%l:ps.1p’’ format described in strftime(3). If the command started less than 7 days ago, the start time is displayed using the ‘%a6.15p’’ format. Otherwise, the start time is displayed using the ‘‘%e%b%y’’ format.

`state` The state is given by a sequence of letters, for example, ‘‘RWNA’’. The first letter indicates the run state of the process:

- `D` Marks a process in disk (or other short term, uninterruptible) wait.
- `I` Marks a process that is idle (sleeping for longer than about 20 seconds).
- `R` Marks a runnable process.
- `S` Marks a process that is sleeping for less than about 20 seconds.
- `T` Marks a stopped process.
- `Z` Marks a dead process (a ‘‘zombie’’).

`Additional characters after these, if any, indicate additional state information:`

- `+` The process is in the foreground process group of its control terminal.
- `<` The process has raised CPU scheduling priority.
- `>` The process has specified a soft limit on memory requirements and is currently exceeding that limit; such a process is (necessarily) not swapped.

	`A`	The process has asked for random page replacement (MADV_RANDOM, from madvise(2), for example, lisp(1) in a garbage collect).
	`E`	The process is trying to exit.
	`L`	The process has pages locked in core (for example, for raw I/O).
	`N`	The process has reduced CPU scheduling priority (see setpriority(2)).
	`S`	The process has asked for FIFO page replacement (MADV_SEQUENTIAL, from madvise(2), for example, a large image processing program using virtual memory to sequentially address voluminous data).
	`s`	The process is a session leader.
	`V`	The process is suspended during a vfork.
	`W`	The process is swapped out.
	`X`	The process is being traced or debugged.
`tt`	An abbreviation for the pathname of the controlling terminal, if any. The abbreviation consists of the three letters following /dev/tty, or, for the console, ‘‘con’’. This is followed by a ‘‘-’’ if the process can no longer reach that controlling terminal (i.e., it has been revoked).	
`wchan`	The event (an address in the system) on which a process waits. When printed numerically, the initial part of the address is trimmed off and the result is printed in hex, for example, 0x80324000 prints as 324000.	

When printing using the command keyword, a process that has exited and has a parent that has not yet waited for the process (in other words, a zombie) is listed as "<defunct>", and a process which is blocked while trying to exit is listed as "<exiting>". Ps makes an educated guess as to the file name and arguments given when the process was created by examining memory or the swap area. The method is inherently somewhat unreliable and in any event a process is entitled to destroy this information, so the names cannot be depended on too much. The ucomm (accounting) keyword can, however, be depended on.

Keywords

The following is a complete list of the available keywords and their meanings. Several of them have aliases (keywords which are synonyms).

`%cpu`	percentage cpu usage (alias pcpu)
`%mem`	percentage memory usage (alias pmem)
`acflag`	accounting flag (alias acflg)
`command`	command and arguments
`cpu`	short-term cpu usage factor (for scheduling)
`flags`	the process flags, in hexadecimal (alias f)
`inblk`	total blocks read (alias inblock)
`jobc`	job control count
`ktrace`	tracing flags
`ktracep`	tracing vnode
`lim`	memoryuse limit
`logname`	login name of user who started the process
`lstart`	time started
`majflt`	total page faults

`minflt`	total page reclaims
`msgrcv`	total messages received (reads from pipes/sockets)
`msgsnd`	total messages sent (writes on pipes/sockets)
`nice`	nice value (alias ni)
`nivcsw`	total involuntary context switches
`nsigs`	total signals taken (alias nsignals)
`nswap`	total swaps in/out
`nvcsw`	total voluntary context switches
`nwchan`	wait channel (as an address)
`oublk`	total blocks written (alias oublock)
`p_ru`	resource usage (valid only for zombie)
`paddr`	swap address
`pagein`	pageins (same as majflt)
`pgid`	process group number
`pid`	process ID
`poip`	pageouts in progress
`ppid`	parent process ID
`pri`	scheduling priority
`re`	core residency time (in seconds; 127 = infinity)
`rgid`	real group ID
`rlink`	reverse link on run queue, or 0
`rss`	resident set size

`rsz`	resident set size + (text size / text use count) (alias rssize)
`rtprio`	realtime priority (101 = not a realtime process)
`ruid`	real user ID
`ruser`	user name (from ruid)
`sess`	session pointer
`sig`	pending signals (alias pending)
`sigcatch`	caught signals (alias caught)
`sigignore`	ignored signals (alias ignored)
`sigmask`	blocked signals (alias blocked)
`sl`	sleep time (in seconds; 127 = infinity)
`start`	time started
`state`	symbolic process state (alias stat)
`svgid`	saved gid from a setgid executable
`svuid`	saved uid from a setuid executable
`tdev`	control terminal device number
`time`	accumulated cpu time, user + system (alias cputime)
`tpgid`	control terminal process group ID
`tsess`	control terminal session pointer
`tsiz`	text size (in Kbytes)
`tt`	control terminal name (two letter abbreviation)
`tty`	full name of control terminal
`uprocp`	process pointer

`ucomm`	name to be used for accounting
`uid`	effective user ID
`upr`	scheduling priority on return from system call (alias usrpri)
`user`	user name (from uid)
`vsz`	virtual size in Kbytes (alias vsize)
`wchan`	wait channel (as a symbolic name)
`xstat`	exit or stop status (valid only for stopped or zombie process)

Files

`/dev`	special files and device names
`/dev/drum`	default swap device
`/dev/kmem`	default kernel memory
`/var/run/dev.db`	/dev name database
`/var/db/kvm_kernel.db`	system namelist database
`/kernel`	default system namelist
`/proc`	the mount point of procfs(5)

See Also

kill(1), w(1), kvm(3), strftime(3), procfs(5), pstat(8)

Bugs

Since ps cannot run faster than the system and is run as any other scheduled process, the information it displays can never be exact.

tcpdump

Name

`tcpdump` - dump traffic on a network

Synopsis

```
tcpdump [ -defglnNOpqStvxX ] [ -c count ] [ -F file ]
[-i interface ] [ -r file ] [ -s snaplen ] [ -T type ]
[ -w file ] [ expression ]
```

Description

Tcpdump prints out the headers of packets on a network interface that match the boolean expression.

Under SunOS with nit or bpf: To run tcpdump you must have read access to /dev/net or /dev/bpf*. Under Solaris with dlpi: You must have read access to the network pseudo device, e.g. /dev/le. Under HP-UX with dlpi: You must be root or it must be installed setuid to root. Under IRIX with snoop: You must be root or it must be installed setuid to root.Under Linux: You must be root or it must be installed setuid to root. Under Ultrix and Digital UNIX: Once the super-user has enabled promiscuous-mode operation using pfconfig(8), any user may run tcpdump. Under BSD: You must have read access to /dev/bpf*.

Options

`-c` Exit after receiving count packets.

`-d`	Dump the compiled packet-matching code in a human readable form to standard output and stop.
`-dd`	Dump packet-matching code as a C program fragment.
`-ddd`	Dump packet-matching code as decimal numbers (preceded with a count).
`-e`	Print the link-level header on each dump line.
`-f`	Print ‘foreign’ internet addresses numerically rather than symbolically (this option is intended to get around serious brain damage in Sun’s yp server -- usually it hangs forever translating nonlocal internet numbers).
`-g`	Should be used with -r and -w flags to convert an IPSO formatted dump that was written earlier using tcpdump, to more portable generic format.
`-F`	Use file as input for the filter expression. An additional expression given on the command line is ignored.
`-i`	Listen on interface. If unspecified, tcpdump searches the system interface list for the lowest numbered, configured up interface (excluding loopback). Ties are broken by choosing the earliest match.
`-l`	Make stdout line buffered. Useful if you want to see the data while capturing it. E.g., ‘‘tcpdump -l | tee dat’’ or ‘‘tcpdump -l> dat & tail -f dat’’.
`-n`	Don’t convert addresses (i.e., host addresses, port numbers, etc.) to names.
`-N`	Don’t print domain name qualification of host names. E.g., if you give this flag then tcpdump will print ‘‘nic’’ instead of ‘‘nic.ddn.mil’’.

`-O`	Do not run the packet-matching code optimizer. This is useful only if you suspect a bug in the optimizer.
`-p`	Don’t put the interface into promiscuous mode. Note that the interface might be in promiscuous mode for some other reason; hence, ‘-p’ cannot be used as an abbreviation for ‘ether host {local-hw-addr} or ether broadcast’.
`-q`	Quick (quiet?) output. Print less protocol information so output lines are shorter.
`-r`	Read packets from file (which was created with the -w option). Standard input is used if file is ‘‘-’’.
`-s`	Snarf snaplen bytes of data from each packet rather than the default of 68 (with SunOS’s NIT, the minimum is actually 96). 68 bytes is adequate for IP, ICMP, TCP and UDP but may truncate protocol information from name server and NFS packets (see below). Packets truncated because of a limited snapshot are indicated in the output with ‘‘[|proto]’’, where proto is the name of the protocol level at which the truncation has occurred. Note that taking larger snapshots both increases the amount of time it takes to process packets and, effectively, decreases the amount of packet buffering. This may cause packets to be lost. You should limit snaplen to the smallest number that will capture the protocol information you’re interested in.
`-T`	Force packets selected by “expression” to be interpreted the specified type. Currently known types are rpc (Remote Procedure Call), rtp (Real-Time Applications protocol), rtcp (Real-Time Applications control protocol), vat (Visual Audio Tool), and wb (distributed White Board).
`-S`	Print absolute, rather than relative, TCP sequence numbers.
`-t`	Don’t print a timestamp on each dump line.
`-tt`	Print an unformatted timestamp on each dump line.

`-v`	(Slightly more) verbose output. For example, the time to live and type of service information in an IP packet is printed.
`-vv`	Even more verbose output. For example, additional fields are printed from NFS reply packets.
`-w`	Write the raw packets to file rather than parsing and printing them out. They can later be printed with the -r option. Standard output is used if file is ‘‘-’’. Use -g flag if the file will be read on some other platform.
`-x`	Print each packet (minus its link level header) in hex. The smaller of the entire packet or snaplen bytes will be printed.
`-X`	Print each packet (minus its link level header) in hex and ascii. The smaller of the entire packet or snaplen bytes will be printed.

Expression

Expression selects which packets will be dumped. If no expression is given, all packets on the net will be dumped. Otherwise, only packets for which expression is ‘true’ will be dumped.

The expression consists of one or more primitives. Primitives usually consist of an id (name or number) preceded by one or more qualifiers. There are three different kinds of qualifier:

type qualifiers say what kind of thing the id name or number refers to. Possible types are host, net and port. E.g., ‘host foo’, ‘net 128.3’, ‘port 20’. If there is no type qualifier, host is assumed.

dir qualifiers specify a particular transfer direction to and/or from id.Possible directions are src, dst, src or dst and src and dst. E.g., ‘src foo’, ‘dst net 128.3’, ‘src or dst port ftp-data’. If there is no dir qualifier, src or dst is assumed. For ‘null’ link layers (i.e. point to point protocols such as slip) the inbound and outbound qualifiers can be used to specify a desired direction.

proto qualifiers restrict the match to a particular protocol. Possible protos are: ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp and udp. E.g., ‘ether src foo’, ‘arp net 128.3’, ‘tcp port 21’. If there is no proto qualifier, all protocols consistent with the type are assumed. E.g., ‘src foo’ means ‘(ip or arp or rarp) src foo’ (except the latter is not legal syntax), ‘net bar’ means ‘(ip or arp or rarp) net bar’ and ‘port 53’ means ‘(tcp or udp) port 53’.

[‘fddi’ is actually an alias for ‘ether’; the parser treats them identically as meaning “the data link level used on the specified network interface.” FDDI headers contain Ethernet-like source and destination addresses, and often contain Ethernet-like packet types, so you can filter on these FDDI fields just as with the analogous Ethernet fields. FDDI headers also contain other fields, but you cannot name them explicitly in a filter expression.]

In addition to the above, there are some special *primitive* keywords that don’t follow the pattern: gateway, broadcast, less, greater and arithmetic expressions. All of these are described below.

More complex filter expressions are built up by using the words and, or and not to combine primitives, for example, ‘host foo and not port ftp and not port ftp-data’. To save typing, identical qualifier lists can be omitted, for example, ‘tcp dst port ftp or ftp-data or domain’ is exactly the same as ‘tcp dst port ftp or tcp dst port ftp-data or tcp dst port domain’.

Allowable primitives are:

`dst host host`	True if the IP destination field of the packet is host, which may be either an address or a name.
`src host host`	True if the IP source field of the packet is host.

`host host`	True if either the IP source or destination of the packet is host. Any of the above host expressions can be prepended with the keywords, ip, arp, or rarp as in: ip host host which is equivalent to: ether proto \ip and host host If host is a name with multiple IP addresses, each address will be checked for a match.
`ether dst ehost T`	rue if the ethernet destination address is ehost. Ehost may be either a name from /etc/ethers or a number (see ethers(3N) for numeric format).
`ether src ehost`	True if the ethernet source address is ehost.
`ether host ehost`	True if either the ethernet source or destination address is ehost.
`gateway host`	True if the packet used host as a gateway. I.e., the ethernet source or destination address was host but neither the IP source nor the IP destination was host. Host must be a name and must be found in both /etc/hosts and /etc/ethers. (An equivalent expression is ether host ehost and not host host which can be used with either names or numbers for host / ehost.)
`dst net net`	True if the IP destination address of the packet has a network number of net. Net may be either a name from /etc/networks or a network number (see networks(4) for details).
`src net net`	True if the IP source address of the packet has a network number of net.
`net net`	True if either the IP source or destination address of the packet has a network number of net.
`net net mask mask`	True if the IP address matches net with the specific netmask. May be qualified with src or dst.

`net net/len`	True if the IP address matches net a netmask len bits wide. May be qualified with src or dst.
`dst port port`	True if the packet is ip/tcp or ip/udp and has a destination port value of port. The port can be a number or a name used in /etc/services (see tcp(4P) and udp(4P)). If a name is used, both the port number and protocol are checked. If a number or ambiguous name is used, only the port number is checked (e.g., dst port 513 will print both tcp/login traffic and udp/who traffic, and port domain will print both tcp/domain and udp/domain traffic).
`src port port`	True if the packet has a source port value of port.
`port port`	True if either the source or destination port of the packet is port. Any of the above port expressions can be prepended with the keywords, tcp or udp, as in: tcp src port port which matches only tcp packets whose source port is port.
`less length`	True if the packet has a length less than or equal to length. This is equivalent to: len <= length.
`greater length`	True if the packet has a length greater than or equal to length. This is equivalent to: len >= length.
`ip proto protocol`	True if the packet is an ip packet (see ip(4P)) of protocol type protocol. Protocol can be a number or one of the names icmp, igrp, udp, nd, or tcp. Note that the identifiers tcp, udp, and icmp are also keywords and must be escaped via backslash (\), which is \\ in the C-shell.
`ether broadcast`	True if the packet is an ethernet broadcast packet. The ether keyword is optional.

`ip broadcast`	True if the packet is an IP broadcast packet. It checks for both the all-zeroes and all-ones broadcast conventions, and looks up the local subnet mask.
`ether multicast`	True if the packet is an ethernet multicast packet. The ether keyword is optional. This is shorthand for ‘ether[0] & 1 != 0’.
`ip multicast`	True if the packet is an IP multicast packet.
`ether proto protocol`	True if the packet is of ether type protocol. Protocol can be a number or a name like ip, arp, or rarp. Note these identifiers are also keywords and must be escaped via backslash (\). [In the case of FDDI (e.g., ‘fddi protocol arp’), the protocol identification comes from the 802.2 Logical Link Control (LLC) header, which is usually layered on top of the FDDI header. Tcpdump assumes, when filtering on the protocol identifier, that all FDDI packets include an LLC header, and that the LLC header is in so-called SNAP format.]
`decnet src host`	True if the DECNET source address is host, which may be an address of the form ‘‘10.123’’, or a DECNET host name. [DECNET host name support is only available on Ultrix systems that are configured to run DECNET.]
`decnet dst host`	True if the DECNET destination address is host.
`decnet host host`	True if either the DECNET source or destination address is host.
`ip, arp, rarp, decnet`	Abbreviations for: ether proto p where p is one of the above protocols.

`lat, moprc, mopdl`	Abbreviations for: ether proto p where p is one of the above protocols. Note that tcpdump does not currently know how to parse these protocols.
`tcp, udp, icmp`	Abbreviations for: ip proto p where p is one of the above protocols.
`expr relop expr`	True if the relation holds, where relop is one of >, <, >=, <=, =, !=, and expr is an arithmetic expression composed of integer constants (expressed in standard C syntax), the normal binary operators [+, -, *, /, &, |], a length operator, and special packet data accessors. To access data inside the packet, use the following syntax: proto [expr : size] Proto is one of ether, fddi, ip, arp, rarp, tcp, udp, or icmp, and indicates the protocol layer for the index operation. The byte offset, relative to the indicated protocol layer, is given by expr. Size is optional and indicates the number of bytes in the field of interest; it can be either one, two, or four, and defaults to one. The length operator, indicated by the keyword len, gives the length of the packet. For example, 'ether[0] & 1 != 0' catches all multicast traffic. The expression 'ip[0] & 0xf != 5' catches all IP packets with options. The expression 'ip[6:2] & 0x1fff = 0' catches only unfragmented datagrams and frag zero of fragmented datagrams. This check is implicitly applied to the tcp and udp index operations. For instance, tcp[0] always means the first byte of the TCP header, and never means the first byte of an intervening fragment.

Primitives may be combined using:

A parenthesized group of primitives and operators (parentheses are special to the Shell and must be escaped).

- Negation (‘!’ or ‘not’).
- Concatenation (‘&&’ or ‘and’).
- Alternation (‘||’ or ‘or’).

Negation has highest precedence. Alternation and concatenation have equal precedence and associate left to right. Note that explicit and tokens, not juxtaposition, are now required for concatenation.

If an identifier is given without a keyword, the most recent keyword is assumed. For example, not host vs and ace is short for not host vs and host ace which should not be confused with not (host vs or ace)

Expression arguments can be passed to tcpdump as either a single argument or as multiple arguments, whichever is more convenient. Generally, if the expression contains Shell metacharacters, it is easier to pass it as a single, quoted argument. Multiple arguments are concatenated with spaces before being parsed.

Examples

To print all packets arriving at or departing from sundown: tcpdump host sundown

To print traffic between helios and either hot or ace: tcpdump host helios and \(hot or ace \)

To print all IP packets between ace and any host except helios: tcpdump ip host ace and not helios

To print all traffic between local hosts and hosts at Berkeley: tcpdump net ucb-ether

To print all ftp traffic through internet gateway snup: (note that the expression is quoted to prevent the shell from (mis-)interpreting the parentheses): tcpdump ‘gateway snup and (port ftp or ftp-data)’

To print traffic neither sourced from nor destined for local hosts (if you gateway to one other net, this stuff should never make it onto your local net): tcpdump ip and not net localnet

To print the start and end packets (the SYN and FIN packets) of each TCP conversation that involves a non-local host: tcpdump ‘tcp[13] & 3 != 0 and not src and dst net localnet’

To print IP packets longer than 576 bytes sent through gateway snup: tcpdump ‘gateway snup and ip[2:2] > 576’

To print IP broadcast or multicast packets that were not sent via ethernet broadcast or multicast: tcpdump ‘ether[0] & 1 = 0 and ip[16] >= 224’

To print all ICMP packets that are not echo requests/replies (i.e., not ping packets): tcpdump ‘icmp[0] != 8 and icmp[0] != 0”

Output Format

The output of tcpdump is protocol dependent. The following gives a brief description and examples of most of the formats.

Link Level Headers

If the ‘-e’ option is given, the link level header is printed out. On Ethernet networks, the source and destination addresses, protocol, and packet length are printed.

On FDDI networks, the ‘-e’ option causes tcpdump to print the ‘frame control’ field, the source and destination addresses, and the packet length. (The ‘frame control’ field governs the interpretation of the rest of the packet. Normal packets (such as those containing IP datagrams) are ‘async’ packets, with a priority value between 0 and 7; for example, ‘async4’. Such packets are assumed to contain an 802.2 Logical Link Control (LLC) packet; the LLC header is printed if it is not an ISO datagram or a so-called SNAP packet.

(N.B.: The following description assumes familiarity with the SLIP compression algorithm described in RFC-1144.)

On SLIP links, a direction indicator (“I” for inbound, “O” for outbound), packet type, and compression information are printed out. The packet type is printed first. The three types are ip, utcp, and ctcp. No further link information is printed for ip packets. For TCP packets, the connection identifier is printed following the type. If the packet is compressed, its encoded header is printed out. The special cases are printed out as *S+n and *SA+n, where n is the amount by which the sequence number (or sequence number and ack) has changed. If it is not a special case, zero or more changes are printed. A change is indicated by U (urgent pointer), W (window), A (ack), S (sequence number), and I (packet ID), followed by a delta (+n or -n), or a new value (=n). Finally, the amount of data in the packet and compressed header length are printed.

For example, the following line shows an outbound compressed TCP packet, with an implicit connection identifier; the ack has changed by 6, the sequence number by 49, and the packet ID by 6; there are 3 bytes of data and 6 bytes of compressed header: O ctcp * A+6 S+49 I+6 3 (6)

ARP/RARP Packets

Arp/rarp output shows the type of request and its arguments. The format is intended to be self explanatory. Here is a short sample taken from the start of an ‘rlogin’ from host rtsg to host csam: arp who-has csam tell rtsg arp reply csam is-at CSAM The first line says that rtsg sent an arp packet asking for the ethernet address of internet host csam. Csam replies with its ethernet address (in this example, ethernet addresses are in caps and internet addresses in lower case).

This would look less redundant if we had done tcpdump -n: arp who-has 128.3.254.6 tell 128.3.254.68 arp reply 128.3.254.6 is-at 02:07:01:00:01:c4

If we had done tcpdump -e, the fact that the first packet is broadcast and the second is point-to-point would be visible: RTSG Broadcast 0806 64: arp who-has csam tell rtsg CSAM RTSG 0806 64: arp reply csam is-at CSAM For the first packet this says the

ethernet source address is RTSG, the destination is the ethernet broadcast address, the type field contained hex 0806 (type ETHER_ARP) and the total length was 64 bytes.

TCP Packets

(N.B.:The following description assumes familiarity with the TCP protocol described in RFC-793. If you are not familiar with the protocol, neither this description nor tcpdump will be of much use to you.)

The general format of a tcp protocol line is: src > dst: flags data-seqno ack window urgent options Src and dst are the source and destination IP addresses and ports. Flags are some combination of S (SYN), F (FIN), P (PUSH) or R (RST) or a single '.' (no flags). Data-seqno describes the portion of sequence space covered by the data in this packet (see example below). Ack is sequence number of the next data expected the other direction on this connection. Window is the number of bytes of receive buffer space available the other direction on this connection. Urg indicates there is 'urgent' data in the packet. Options are tcp options enclosed in angle brackets (e.g., <mss 1024>).

Src, dst and flags are always present. The other fields depend on the contents of the packet's tcp protocol header and are output only if appropriate.

Here is the opening portion of an rlogin from host rtsg to host csam.

```
rtsg.1023 > csam.login: S 768512:768512(0) win 4096 <mss 1024>
csam.login > rtsg.1023: S 947648:947648(0) ack 768513 win 4096 <mss1024>
rtsg.1023 > csam.login: . ack 1 win 4096
rtsg.1023 > csam.login: P 1:2(1) ack 1 win 4096
csam.login > rtsg.1023: . ack 2 win 4096
rtsg.1023 > csam.login: P 2:21(19) ack 1 win 4096
csam.login > rtsg.1023: P 1:2(1) ack 21 win 4077
csam.login > rtsg.1023: P 2:3(1) ack 21 win 4077 urg 1
csam.login > rtsg.1023: P 3:4(1) ack 21 win 4077 urg 1
```

The first line says that tcp port 1023 on rtsg sent a packet to port login on csam. The S indicates that the SYN flag was set. The packet sequence number was 768512 and it contained no data. (The notation is ‘first:last(nbytes)’ which means ‘sequence numbers first up to but not including last which is nbytes bytes of user data’.) There was no piggy-backed ack, the available receive window was 4096 bytes and there was a max-segment- size option requesting an mss of 1024 bytes.

Csam replies with a similar packet except it includes a piggy-backed ack for rtsg’s SYN. Rtsg then acks csam’s SYN. The ‘.’ means no flags were set. The packet contained no data so there is no data sequence number. Note that the ack sequence number is a small integer (1). The first time tcpdump sees a tcp ‘conversation’, it prints the sequence number from the packet. On subsequent packets of the conversation, the difference between the current packet’s sequence number and this initial sequence number is printed. This means that sequence numbers after the first can be interpreted as relative byte positions in the conversation’s data stream (with the first data byte each direction being ‘1’). ‘-S’ will override this feature, causing the original sequence numbers to be output.

On the 6th line, rtsg sends csam 19 bytes of data (bytes 2 through 20 in the rtsg -> csam side of the conversation). The PUSH flag is set in the packet. On the 7th line, csam says it’s received data sent by rtsg up to but not including byte 21. Most of this data is apparently sitting in the socket buffer since csam’s receive window has gotten 19 bytes smaller. Csam also sends one byte of data to rtsg in this packet. On the 8th and 9th lines, csam sends two bytes of urgent, pushed data to rtsg.

If the snapshot was small enough that tcpdump didn’t capture the full TCP header, it interprets as much of the header as it can and then reports “[|tcp]” to indicate the remainder could not be interpreted. If the header contains a bogus option (one with a length that’s either too small or beyond the end of the header), tcpdump reports it as “[bad opt]” and does not interpret any further options (since it’s impossible to tell where they start). If the header length indicates options are present but the IP datagram length is not long enough for the options to actually be there, tcpdump reports it as “[bad hdr length].”

UDP Packets

UDP format is illustrated by this rwho packet: actinide.who > broadcast.who: udp 84 This says that port who on host actinide sent a udp datagram to port who on host broadcast, the Internet broadcast address. The packet contained 84 bytes of user data.

Some UDP services are recognized (from the source or destination port number) and the higher level protocol information printed. In particular, Domain Name service requests (RFC-1034/1035) and Sun RPC calls (RFC-1050) to NFS.

UDP Name Server Requests

(N.B.:The following description assumes familiarity with the Domain Service protocol described in RFC-1035. If you are not familiar with the protocol, the following description will appear to be written in greek.)

Name server requests are formatted as src > dst: id op? flags qtype qclass name (len) h2opolo.1538 > helios.domain: 3+ A? ucbvax.berkeley.edu. (37) Host h2opolo asked the domain server on helios for an address record (qtype=A) associated with the name ucb- vax.berkeley.edu. The query id was ‘3’. The ‘+’ indicates the recursion desired flag was set. The query length was 37 bytes, not including the UDP and IP protocol headers. The query operation was the normal one, Query, so the op field was omitted. If the op had been anything else, it would have been printed between the ‘3’ and the ‘+’. Similarly, the qclass was the normal one, C_IN, and omitted. Any other qclass would have been printed immediately after the ‘A’.

A few anomalies are checked and may result in extra fields enclosed in square brackets:If a query contains an answer, name server or authority section, ancount, nscount, or arcount are printed as ‘[na]’, ‘[nn]’ or ‘[nau]’ where n is the appropriate count. If any of the response bits are set (AA, RA or rcode) or any of the ‘must be zero’ bits are set in bytes two and three,‘[b2&3=x]’ is printed, where x is the hex value of header bytes two and three.

UDP Name Server Responses

Name server responses are formatted as

src > dst: id op rcode flags a/n/au type class data (len)
helios.domain > h2opolo.1538: 3 3/3/7 A 128.32.137.3 (273)
helios.domain > h2opolo.1537: 2 NXDomain* 0/1/0 (97)

In the first example, helios responds to query id 3 from h2opolo with 3 answer records, 3 name server records and 7 authority records. The first answer record is type A (address) and its data is internet address 128.32.137.3. The total size of the response was 273 bytes, excluding UDP and IP headers. The op (Query) and response code (NoError) were omitted, as was the class (C_IN) of the A record.

In the second example, helios responds to query 2 with a response code of non-existent domain (NXDomain) with no answers, one name server and no authority records. The '*' indicates that the authoritative answer bit was set. Since there were no answers, no type, class or data were printed.

Other flag characters that might appear are '-' (recursion available, RA, not set) and '|' (truncated message, TC, set). If the 'question' section doesn't contain exactly one entry, '[nq]' is printed.

Note that name server requests and responses tend to be large and the default snaplen of 68 bytes may not capture enough of the packet to print. Use the -s flag to increase the snaplen if you need to seriously investigate name server traffic. '-s 128' has worked well for me.

NFS Requests and Replies

Sun NFS (Network File System) requests and replies are printed as:

src.xid > dst.nfs: len op args
src.nfs > dst.xid: reply stat len op results
sushi.6709 > wrl.nfs: 112 readlink fh 21,24/10.73165
wrl.nfs > sushi.6709: reply ok 40 readlink "../var"
sushi.201b > wrl.nfs: 144 lookup fh 9,74/4096.6878 "xcolors"
wrl.nfs > sushi.201b: reply ok 128 lookup fh 9,74/4134.3150

In the first line, host sushi sends a transaction with id 6709 to wrl (note that the number following the src host is a transaction id, not the source port). The request was 112 bytes, excluding the UDP and IP headers. The operation was a readlink (read symbolic link) on file handle (fh) 21,24/10.731657119. (If one is lucky, as in this case, the file handle can be interpreted as a major and minor device number pair, followed by the inode number and generation number.) Wrl replies ‘ok’ with the contents of the link.

In the third line, sushi asks wrl to lookup the name ‘xcolors’ in directory file 9,74/4096.6878. Note that the data printed depends on the operation type. The format is intended to be self explanatory if read in conjunction with an NFS protocol spec.

If the -v (verbose) flag is given, additional information is printed. For example:

sushi.1372a > wrl.nfs: 148 read fh 21,11/12.195 8192 bytes @ 24576
wrl.nfs > sushi.1372a: reply ok 1472 read REG 100664 ids 417/0 sz 29388

(-v also prints the IP header TTL, ID, and fragmentation fields, which have been omitted from this example.) In the first line, sushi asks wrl to read 8192 bytes from file 21,11/12.195, at byte offset 24576. Wrl replies ‘ok’; the packet shown on the second line is the first fragment of the reply, and hence is only 1472 bytes long (the other bytes will follow in subsequent fragments, but these fragments do not have NFS or even UDP headers and so might not be printed, depending on the filter expression used). Because the -v flag is given, some of the file attributes (which are returned in addition to the file data) are printed: the file type (“REG”, for regular file), the file mode (in octal), the uid and gid, and the file size.

If the -v flag is given more than once, even more details are printed.

NOTE: that NFS requests are very large and much of the detail won’t be printed unless snaplen is increased. Try using ‘-s 192’ to watch NFS traffic.

NFS reply packets do not explicitly identify the RPC operation. Instead, tcpdump keeps track of “recent” requests, and matches them to the replies using the transaction ID. If a reply does not closely follow the corresponding request, it might not be parsable.

KIP Appletalk (DDP in UDP)

Appletalk DDP packets encapsulated in UDP datagrams are de-encapsulated and dumped as DDP packets (i.e., all the UDP header information is discarded). The file /etc/atalk.names is used to translate appletalk net and node numbers to names. Lines in this file have the form number name:

1.254 ether
16.1 icsd-net
1.254.110 ace

The first two lines give the names of appletalk networks. The third line gives the name of a particular host (a host is distinguished from a net by the 3rd octet in the number - a net number must have two octets and a host number must have three octets.) The number and name should be separated by whitespace (blanks or tabs). The /etc/atalk.names file may contain blank lines or comment lines (lines starting with a '#').

Appletalk addresses are printed in the form net.host.port

144.1.209.2 > icsd-net.112.220
office.2 > icsd-net.112.220
jssmag.149.235 > icsd-net.2

(If the /etc/atalk.names doesn't exist or doesn't contain an entry for some appletalk host/net number, addresses are printed in numeric form.) In the first example, NBP (DDP port 2) on net 144.1 node 209 is sending to whatever is listening on port 220 of net icsd node 112. The second line is the same except the full name of the source node is known ('office'). The third line is a send from port 235 on net jssmag node 149 to broadcast on the icsd-net NBP port (note that the broadcast address (255) is indicated by a net name with no host number - for this reason it's a good idea to keep node names and net names distinct in /etc/atalk.names).

NBP (name binding protocol) and ATP (Appletalk transaction protocol) packets have their contents interpreted. Other protocols just dump the protocol name (or number if no name is registered for the protocol) and packet size.

NBP packets are formatted like the following examples:

icsd-net.112.220 > jssmag.2: nbp-lkup 190: “=:LaserWriter@*”
jssmag.209.2 > icsd-net.112.220: nbp-reply 190: “RM1140:LaserWriter@*” 250
techpit.2 > icsd-net.112.220: nbp-reply 190: “techpit:LaserWriter@*” 186

The first line is a name lookup request for laserwriters sent by net icsd host 112 and broadcast on net jssmag. The nbp id for the lookup is 190. The second line shows a reply for this request (note that it has the same id) from host jssmag.209 saying that it has a laserwriter resource named “RM1140” registered on port 250. The third line is another reply to the same request saying host techpit has laserwriter “techpit” registered on port 186.

ATP packet formatting is demonstrated by the following example:

jssmag.209.165 > helios.132: atp-req 12266<0-7> 0xae030001
helios.132 > jssmag.209.165: atp-resp 12266:0 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:1 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:2 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:3 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:4 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:5 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:6 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp*12266:7 (512) 0xae040000
jssmag.209.165 > helios.132: atp-req 12266<3,5> 0xae030001
helios.132 > jssmag.209.165: atp-resp 12266:3 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:5 (512) 0xae040000
jssmag.209.165 > helios.132: atp-rel 12266<0-7> 0xae030001
jssmag.209.133 > helios.132: atp-req* 12267<0-7> 0xae030002

Jssmag.209 initiates transaction id 12266 with host helios by requesting up to 8 packets (the ‘<0-7>’). The hex number at the end of the line is the value of the ‘userdata’ field in the request.

Helios responds with 8 512-byte packets. The ‘:digit’ following the transaction id gives the packet sequence number in the transaction and the number in parentheses is the amount of data in the packet, excluding the atp header. The ‘*’ on packet 7 indicates that the EOM bit was set.

Jssmag.209 then requests that packets 3 & 5 be retransmitted. Helios resends them then jssmag.209 releases the transaction. Finally,jssmag.209 initiates the next request. The ‘*’ on the request indicates that XO (‘exactly once’) was not set.

IP Fragmentation

Fragmented Internet datagrams are printed as

(frag id:size@offset+)
(frag id:size@offset)

(The first form indicates there are more fragments. The second indicates this is the last fragment.)

Id is the fragment id. Size is the fragment size (in bytes) excluding the IP header. Offset is this fragment’s offset (in bytes) in the original datagram.

The fragment information is output for each fragment. The first fragment contains the higher level protocol header and the frag info is printed after the protocol info. Fragments after the first contain no higher level protocol header and the frag info is printed after the source and destination addresses. For example, here is part of an ftp from arizona.edu to lbl-rtsg.arpa over a CSNET connection that doesn’t appear to handle 576 byte datagrams:

arizona.ftp-data > rtsg.1170: . 1024:1332(308) ack 1 win 4096 (frag 595a:328@0+)
arizona > rtsg: (frag 595a:204@328)
rtsg.1170 > arizona.ftp-data: . ack 1536 win 2560

There are a couple of things to note here: First, addresses in the 2nd line don’t include port numbers. This is because the TCP protocol information is all in the first fragment and we have no idea what the port or sequence numbers are when we print the later

fragments. Second, the tcp sequence information in the first line is printed as if there were 308 bytes of user data when, in fact, there are 512 bytes (308 in the first frag and 204 in the second). If you are looking for holes in the sequence space or trying to match up acks with packets, this can fool you.

A packet with the IP don't fragment flag is marked with a trailing (DF).

Timestamps

By default, all output lines are preceded by a timestamp. The timestamp is the current clock time in the form hh:mm:ss.frac and is as accurate as the kernel's clock. The timestamp reflects the time the kernel first saw the packet. No attempt is made to account for the time lag between when the ethernet interface removed the packet from the wire and when the kernel serviced the 'new packet' interrupt.

See Also

traffic(1C), nit(4P), bpf(4), pcap(3)

Authors

Van Jacobson, Craig Leres and Steven McCanne, all of the Lawrence Berkeley National Laboratory, University of California, Berkeley, CA.

The current version is available via anonymous ftp:

ftp://ftp.ee.lbl.gov/tcpdump.tar.Z

Bugs

Please send bug reports to tcpdump@ee.lbl.gov.

NIT doesn't let you watch your own outbound traffic, BPF will. We recommend that you use the latter.

Some attempt should be made to reassemble IP fragments or, at least to compute the right length for the higher level protocol.

Name server inverse queries are not dumped correctly: The (empty) question section is printed rather than real query in the answer section.Some believe that inverse queries are themselves a bug and prefer to fix the program generating them rather than tcpdump.

Apple Ethertalk DDP packets could be dumped as easily as KIP DDP packets but aren't. Even if we were inclined to do anything to promote the use of Ethertalk (we aren't), LBL doesn't allow Ethertalk on any of its networks so we'd would have no way of testing this code.

A packet trace that crosses a daylight savings time change will give skewed time stamps (the time change is ignored).

Filters expressions that manipulate FDDI headers assume that all FDDI packets are encapsulated Ethernet packets. This is true for IP, ARP, and DECNET Phase IV, but is not true for protocols such as ISO CLNS. Therefore, the filter may inadvertently accept certain packets that do not properly match the filter expression.

traceroute

Name

`traceroute` - print the route packets take to network host

Synopsis

```
traceroute [ -dnrv ] [ -g gw_host ] [ -m max_ttl ] [ -p port ]
[ -q nqueries ] [ -s src_addr ] [ -t tos ] [ -w waittime ]
host [ packetlen ]
```

Description

The Internet is a large and complex aggregation of network hardware, connected together by gateways. Tracking the route one's packets follow (or finding the miscreant gateway that's discarding your packets) can be difficult. Traceroute utilizes the IP protocol 'time to live' field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host.

The only mandatory parameter is the destination host name or IP number. The default probe datagram length is 40 bytes, but this may be increased by specifying a packet length (in bytes) after the destination host name.

Other options are:

`-g`	Specify a loose source route gateway (8 maximum).
`-m`	Set the max time-to-live (max number of hops) used in outgoing probe packets. The default is 30 hops (the same default used for TCP connections).

`-n` Print hop addresses numerically rather than symbolically and numerically (saves a nameserver address-to-name lookup for each gateway found on the path).

`-p` Set the base UDP port number used in probes (default is 33434). Traceroute hopes that nothing is listening on UDP ports base to base + nhops - 1 at the destination host (so an ICMP PORT_UNREACHABLE message will be returned to terminate the route tracing). If something is listening on a port in the default range, this option can be used to pick an unused port range.

`-r` Bypass the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it (e.g., after the interface was dropped by routed(8C)).

`-s` Use the following IP address (which must be given as an IP number, not a hostname) as the source address in outgoing probe packets. On hosts with more than one IP address, this option can be used to force the source address to be something other than the IP address of the interface the probe packet is sent on. If the IP address is not one of this machine's interface addresses, an error is returned and nothing is sent.

`-t` Set the type-of-service in probe packets to the following value (default zero). The value must be a decimal integer in the range 0 to 255. This option can be used to see if different types-of-service result in different paths. (If you are not running 4.4bsd, this may be academic since the normal network services like telnet and ftp don't let you control the TOS). Not all values of TOS are legal or meaningful - see the IP spec for definitions. Useful values are probably '-t 16' (low delay) and '-t 8' (high throughput).

`-v` Verbose output. Received ICMP packets other than TIME_EXCEEDED and UNREACHABLEs are listed.

`-w` Set the time (in seconds) to wait for a response to a probe (default 5 sec.).

This program attempts to trace the route an IP packet would follow to some internet host by launching UDP probe packets with a small ttl (time to live) then listening for an ICMP “time exceeded” reply from a gateway. We start our probes with a ttl of one and increase by one until we get an ICMP “port unreachable” (which means we got to “host”) or hit a max (which defaults to 30 hops & can be changed with the -m flag). Three probes (change with -q flag) are sent at each ttl setting and a line is printed showing the ttl, address of the gateway and round trip time of each probe. If the probe answers come from different gateways, the address of each responding system will be printed. If there is no response within a 5 sec. timeout interval (changed with the -w flag), a “*” is printed for that probe.

We don’t want the destination host to process the UDP probe packets so the destination port is set to an unlikely value (if some clod on the destination is using that value, it can be changed with the -p flag).

A sample use and output might be:

```
[yak 71]% traceroute nis.nsf.net.
traceroute to nis.nsf.net (35.1.1.48), 30 hops max, 38 byte packet
    1 helios.ee.lbl.gov (128.3.112.1) 19 ms 19 ms 0 ms
    2 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 39 ms 19 ms
    3 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 39 ms 19 ms
    4 ccngw-ner-cc.Berkeley.EDU (128.32.136.23) 39 ms 40 ms 39 ms
    5 ccn-nerif22.Berkeley.EDU (128.32.168.22) 39 ms 39 ms 39 ms
    6 128.32.197.4 (128.32.197.4) 40 ms 59 ms 59 ms
    7 131.119.2.5 (131.119.2.5) 59 ms 59 ms 59 ms
    8 129.140.70.13 (129.140.70.13) 99 ms 99 ms 80 ms
    9 129.140.71.6 (129.140.71.6) 139 ms 239 ms 319 ms
    10 129.140.81.7 (129.140.81.7) 220 ms 199 ms 199 ms
    11 nic.merit.edu (35.1.1.48) 239 ms 239 ms 239 ms
```

Note that lines 2 & 3 are the same. This is due to a buggy kernel on the 2nd hop system - lbl-csam.arpa - that forwards packets with a zero ttl (a bug in the distributed version of 4.3BSD). Note that you have to guess what path the packets are taking cross-country since the NSFNet (129.140) does not supply address-to-name translations for its NSSes.

A more interesting example is:

```
[yak 72]% traceroute allspice.lcs.mit.edu.
traceroute to allspice.lcs.mit.edu (18.26.0.115), 30 hops max
    1 helios.ee.lbl.gov (128.3.112.1) 0 ms 0 ms 0 ms
    2 lilac-dmc.Berkeley.EDU (128.32.216.1) 19 ms 19 ms 19 ms
    3 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 19 ms 19 ms
    4 ccngw-ner-cc.Berkeley.EDU (128.32.136.23) 19 ms 39 ms 39 ms
    5 ccn-nerif22.Berkeley.EDU (128.32.168.22) 20 ms 39 ms 39 ms
    6 128.32.197.4 (128.32.197.4) 59 ms 119 ms 39 ms
    7 131.119.2.5 (131.119.2.5) 59 ms 59 ms 39 ms
    8 129.140.70.13 (129.140.70.13) 80 ms 79 ms 99 ms
    9 129.140.71.6 (129.140.71.6) 139 ms 139 ms 159 ms
    10 129.140.81.7 (129.140.81.7) 199 ms 180 ms 300 ms
    11 129.140.72.17 (129.140.72.17) 300 ms 239 ms 239 ms
    12 * * *
    13 128.121.54.72 (128.121.54.72) 259 ms 499 ms 279 ms
    14 * * *
    15 * * *
    16 * * *
    17 * * *
    18 ALLSPICE.LCS.MIT.EDU (18.26.0.115) 339 ms 279 ms 279 ms
```

Note that the gateways 12, 14, 15, 16 & 17 hops away either do not send ICMP “time exceeded” messages or send them with a ttl too small to reach us. The 14 - 17 gateways are running the MIT C Gateway code that does not send time exceeded messages. It is unclear what happens with gateway 12.

The silent gateway 12 in the above example may be the result of a bug in the 4.[23]BSD network code (and its derivatives): 4.x (x<= 3) sends an unreachable message using whatever ttl remains in the original datagram. Since, for gateways, the remaining ttl is zero, the ICMP “time exceeded” is guaranteed to not make it back to us. The behavior of this bug is slightly more interesting when it appears on the destination system:

```
1 helios.ee.lbl.gov (128.3.112.1) 0 ms 0 ms 0 ms
2 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 19 ms 39 ms
3 lilac-dmc.Berkeley.EDU (128.32.216.1) 19 ms 39 ms 19 ms
4 ccngw-ner-cc.Berkeley.EDU (128.32.136.23) 39 ms 40 ms 19 ms
5 ccn-nerif35.Berkeley.EDU (128.32.168.35) 39 ms 39 ms 39 ms
6 csgw.Berkeley.EDU (128.32.133.254) 39 ms 59 ms 39 ms
7 * * *
8 * * *
9 * * *
10 * * *
11 * * *
12 * * *
13 rip.Berkeley.EDU (128.32.131.22) 59 ms ! 39 ms ! 39 ms !
```

Notice that there are 12 “gateways” (13 is the final destination) and exactly the last half of them are “missing”. What’s really happening is that rip (a Sun-3 running Sun OS3.5) is using the ttl from our arriving datagram as the ttl in its ICMP reply. So, the reply will time out on the return path (with no notice sent to anyone since ICMP’s aren’t sent for ICMP’s) until we probe with a ttl that’s at least twice the path length. I.e., rip is really only 7 hops away. A reply that returns with a ttl of 1 is a clue this problem exists. Traceroute prints a “!” after the time if the ttl is <= 1. Since vendors ship a lot of obsolete (DEC’s Ultrix, Sun 3.x) or non-standard (HPUX) software, expect to see this problem frequently and/or take care picking the target host of your probes.

Other possible annotations after the time are !H, !N, or !P (got a host, network or protocol unreachable, respectively), !S or !F (source route failed or fragmentation needed - neither of these should ever occur and the associated gateway is busted if you see one), !X (communication administratively prohibited), or !<N> (ICMP unreach-

able code N).If almost all the probes result in some kind of unreachable, traceroute will give up and exit.

This program is intended for use in network testing, measurement and management. It should be used primarily for manual fault isolation. Because of the load it could impose on the network, it is unwise to use traceroute during normal operations or from automated scripts.

See Also

netstat(1), ping(8)

Author

Implemented by Van Jacobson from a suggestion by Steve Deering. Debugged by a cast of thousands with particularly cogent suggestions or fixes from C. Philip Wood, Tim Seaver and Ken Adelman.

The current version is available via anonymous ftp:

ftp://ftp.ee.lbl.gov/traceroute.tar.Z

Bugs

Please send bug reports to traceroute@ee.lbl.gov.

uptime

Name

`uptime` - show how long system has been running

Synopsis

```
uptime
```

Description

The uptime utility displays the current time, the length of time the system has been up, the number of users, and the load average of the system over the last 1, 5, and 15 minutes.

Files

/kernel system name list

See Also

w(1)

History

The uptime command appeared in 3.0BSD.

vmstat

Name

vmstat - report virtual memory statistics

Synopsis

vmstat [-ims] [-c count] [-M core] [-N system] [-w wait] [if,pass] [disks]

Description

Vmstat reports certain kernel statistics kept about process, virtual memory, disk, trap and CPU activity.

The options are as follows:

`c`	Repeat the display count times. The first display is for the time since a reboot and each subsequent report is for the time period since the last display. If no repeat count is specified, and -w is specified, the default is infinity, otherwise the default is one.
`-i`	Report on the number of interrupts taken by each device since system startup.
`-M`	Extract values associated with the name list from the specified core instead of the default /dev/kmem.
`-N`	Extract the name list from the specified system instead of the default / kernel.

`-m` Report on the usage of kernel dynamic memory listed first by size of allocation and then by type of usage.

.

`-s` Display the contents of the sum structure, giving the total number of several kinds of paging related events which have occurred since system startup.

`-w` Pause wait seconds between each display. If no repeat wait interval is specified, the default is 1 second.

By default, vmstat displays the following information:

`procs` Information about the numbers of processes in various states.

r	in run queue
b	blocked for resources (i/o, paging, etc.)
w	runnable or short sleeper (< 20 secs) but swapped

`memory` Information about the usage of virtual and real memory. Virtual pages (reported in units of 1024 bytes) are considered active if they belong to processes which are running or have run in the last 20 seconds.

`avm` active virtual pages

`fre` size of the free list

`page` Information about page faults and paging activity. These are averaged each five seconds, and given in units per second.

flt	total number of page faults
re	page reclaims (simulating reference bits)
pi	pages paged in
po	pages paged out
fr	pages freed per second
sr	pages scanned by clock algorithm, per-second

`disks` Disk operations per second (this field is system dependent). Typically paging will be split across the available drives. The header of the field is the first two characters of the disk name and the unit number. If more than three disk drives are configured in the system, vmstat displays only the first three drives, unless the user specifies the -n argument to increase the number of drives displayed. This will probably cause the display to exceed 80 columns, however. To force vmstat to display specific drives, their names may be supplied on the command line. vmstat defaults to show disks first, and then various other random devices in the system to add up to three devices, if there are that many devices in the system. If devices are specified on the command line, or if a device type matching pattern is specified (see above), vmstat will only display the given devices or the devices matching the pattern, and will not randomly select other devices in the system.

`faults` Trap/interrupt rate averages per second over last 5 seconds.

in	device interrupts per interval (including clock interrupts)
sy	system calls per interval
cs	cpu context switch rate (switches/interval)

`cpu` Breakdown of percentage usage of CPU time.

us	user time for normal and low priority processes
sy	system time
id	cpu idle

Examples

The command: `vmstat -w 5` will print what the system is doing every five seconds; this is a good choice of printing interval since this is how often some of the statistics are sampled in the system. Others vary every second and running the output for a while will make it apparent which are recomputed every second.

The command: `vmstat -p da -p cd -w 1` will tell vmstat to select the first three direct access or CDROM devices and display statistics on those devices, as well as other systems statistics every second.

Files

/kernel default kernel namelist /dev/kmem default memory file

See Also

fstat(1), netstat(1), nfsstat(1), ps(1), systat(1), iostat(8), pstat(8)

The sections starting with “Interpreting system activity” in Installing and Operating 4.3BSD.

Bugs

The -c and -w options are only available with the default output.

List of Commands

Introducing the Command-Line Interface

Environment Commands

Transaction Mode

General CLI Features

Saving Configuration Changes

Interface Commands

General Commands

Viewing All Interfaces

Deleting Any Logical Interface

Viewing Tunnels

Viewing Status and Statistics

ARP

ARP Commands

ATM Interfaces

Physical ATM Interfaces

Logical ATM Interfaces

Ethernet Interfaces

Physical Ethernet Interfaces

Logical Ethernet Interfaces

Transparent Mode

Configuring Transparent Mode

Link Aggregation

Configuring Link Aggregation

lacp_mode
lacp_timer
min_ports
port_priority

system_priority
txpolicy

Link Redundancy

Configuring Link Redundancy

```
add linkredundancy ..... 71
      group <1—1024>
            port phys_if_name
delete linkredundancy ..... 71
      group <1—1024>
            port phys_if_name
show ..... 72
      linkredundancy
            groups
            group <1—1024>
```

Point-to-Point Over Ethernet

Configuring Profiles

```
add pppoe profile name profile_name interface phys_if_name mode <connect-
 on-demand | keep-alive> noauth ..... 72
      timeout <30-259200; 300, 60>
      peername name
      description name
      mss mss_value
      mtu <136-1492>
add pppoe profile name profile_name interface phys_if_name mode mode_name
 username name password password ..... 73
      authtype PAP | CHAP CASE
      timeout <30-259200; 300, 60><
      peername name
      description name
      mss mss_value
      mtu <136-1492>
```

Configuring PPPoE Logical Interface

Configuring PPPoE Physical Interface

FDDI Interfaces

Physical FDDI Interfaces

Logical FDDI Interfaces

ISDN Interfaces

Physical ISDN Interfaces

Logical ISDN Interfaces

Loopback Interfaces

Logical Loopback Interfaces

Logical or Physical Loopback Interfaces

Modem Interfaces

Serial Interfaces

Physical Serial Interfaces

Logical Serial Interfaces

VPP Interfaces

Create Appropriate Static Routes

VPP Interface Commands

System Configuration Commands

System Configuration Summary

Configuring Banner and Login Messages

Configuring DHCP

DHCP Service Commands

Configuring DHCP Clients

Configuring DHCP Servers

Backup and Restore Files

Manually Backing Up

Scheduling Backups

Transferring Backup Files to a Remote Server

Restore Files from Locally Stored Backup Files

Restore Files from Backup Files Stored on Remote Server

Show Backup Commands

```
            minute
            status
```

Schedule Jobs Through Crontab File

Scheduling Jobs

```
set cron..................................................... 149
      job name command name
      job name command name timezone <local | utc> dayofmonth <1-31>
      job name command name timezone <local | utc> dayofweek <0-7>
      job name command name timezone <local | utc> hour <0-23>
      job name command name timezone <local | utc> minute <0-59>
      job name on
      mailto email_addr
```

Adding Jobs

```
add cron job name command name timezone <local | utc> ....... 149
      dayofmonth <1-31>
            hour <0-23>
            minute <0-59>
      dayofweek <0-7>
            hour <0-23>
            minute <0-59>
      mailto email_addr
```

Deleting Jobs

```
delete cron.................................................. 150
      job name
      job name dayofmonth <1-31>
      job name dayofweek <0-7>
      mailto email_addr
```

Show Cron Commands

```
show cron.................................................... 151
      job name command
      job name dayofmonth
      job name dayofweek
      job name hour
```

System Failure Notification Configuration

Enabling System Failure Notification

Show System Failure Notification

DNS

Setting DNS

Show DNS

Deleting DNS

Static Host Address Assignment Configuration

Adding New Host Names

Modifying Host Names

Deleting Host Names

Showing Host Names

Host Name Configuration

Managing IPSO Images

Show IPSO Images

Deleting IPSO Images

Test Boot, Reboot, and Halt IPSO Images

Downloading IPSO Images

Managing Configuration Sets

Configuration Set Commands

Mail Relay Configuration

Mail Relay Commands

System Logging Configuration

Logging Commands (Systems with Disks)

Logging Commands (Flash-Based Systems)

Optional Disk Configuration (Flash-Based Systems)

Configuring an Optional Disk

Core-Dump Server Configuration (Flash-Based Systems)

Sending Core Files to a Remote Server

Date and Time Configuration

Setting Date and Time from Server

Setting Date and Time Manually

Show Date and Clock Commands

Configuring Daylight Savings Rules

Disk Commands

Viewing Disk Information

Disk Mirroring Commands

Configuring Disk Mirroring

NTP

Configuring NTP

Package Commands

Managing Packages

Advanced System Tuning Commands

Controlling Sequence Validation

Tuning the TCP/IP Stack

Router Alert IP Option

IP1260 Port Optimization

High Availability Commands

External Load Balancer Command

VRRP Commands

General VRRP Commands

Simplified Method Monitored-Circuit VRRP

Full Method Monitored-Circuit VRRP

VRRPv2

```
      priority <1-254>
      vmac-mode <default-vmac|extended-vmac|interface-vmac|static-
      vmac mac_address>
```

IP Clustering Commands

General Clustering Commands

Clustering Administration

Managing Join-Time Shared Features

Configuring Join-Time Shared Features

Installing IPSO Images on a Cluster

SNMP Commands

SNMP Description

SNMP Command Set

Enabling/Disabling and Setting SNMP

Enabling and Disabling SNMP Traps

IPv6 Commands

Configuration Summary

Interface Commands

Neighbor Discovery Protocol

Tunnels

IPv6 to IPv4

IPv6 Over IPv4

IPv6 Routing Configuration

RIPng

```
      packets
      errors
      neighbors
      summary
```

Route Aggregation

```
set ipv6 aggregate ip6_prefix ..... 263
      off
      contributing-protocol
      <all | direct | static | aggregate | ripng> off
      contributing-protocol
      <all | direct | static | aggregate | ripng>
      contributing-route <all | ip6_prefix> <on | off>
```

Static Routes

```
set ipv6 static-route ..... 264
      ip6_prefix nexthop gateway ip6_address priority <1-8> <on | off>
      default nexthop gateway ip6_address priority < 1-8> <on | off>
      ip6_prefix nexthop gateway ip6_address interface if_name priority
      <1-8> <on | off>
      default nexthop gateway ip6_address interface if_name <on | off>
      ip6_prefix nexthop reject
      default nexthop reject
      ip6_prefix nexthop blackhole
      default nexthop blackhole
      ip6_prefix off
      default off
```

ICMP Router Discovery

```
set ipv6 rdisc6 interface if_name..... 267
      <on | off>
      min-adv-interval <3-1800>
      min-adv-interval default
      max-adv-interval <4-1800>
      max-adv-interval default
      hop-limit <0–255>
      hop-limit default
      managed-config <on |off>
```

VRRP for IPv6

Show Routing Summary Commands

Host Name Configuration

Network Access and Services

Network Security and Access Commands

Network Access and Services

Licenses

Configuring Software Licenses

IPsec Commands (IPSO Implementation)

General IPsec Commands

Proposal Commands

Filter Commands

Certificate Commands

Policy Commands

```
            life-sec <0—700000>
            life-mb <0—65000>
            ike-group <1 | 2 | 5>
            pfs-group <1 | 2 | 5 | none>
      x509cert name
            life-sec <0—700000>
            life-mb <0—65000>
            ike-group <1 | 2 | 5>
            pfs-group <1 | 2 | 5 | none>
set ipsec policy name..... 303
      proposal name priority integer
      psk secret_key
      x509cert name
      life-sec <0—700000>
      life-mb <0—65000>
      ike-group <1 | 2 | 5>
      pfs-group <1 | 2 | 5 | none>
show ipsec policy..... 305
      all
      name
delete ipsec policy..... 305
      all
      name
            proposal name
```

Rule Commands

```
add ipsec rule name mode tunnel..... 306
      local-address ip_address remote-address ip_address
            policy name
            src-filter name
            dst-filter name
            inc-end-points <on | off>
            logical-interface <on | off>
            hello-prot <on | off> [hello-inv <0—21666> dead-inv <0—
            65000>]
      local-address6 ip6_address remote-address6 ip6_address
            policy name
```

Miscellaneous IPsec Commands

AAA

Viewing AAA Configuration

Configuring Service Modules

Configuring Service Profiles

Configuring Authentication Profiles

Configuring Account Profiles

Configuring Session Profiles

Configuring RADIUS

Configuring TACPLUS

SSH

Enabling/Disabling SSH Service

Configuring Server Options

Configuring and Managing SSH Key Pairs

Voyager Web Access (SSL)

Enabling SSL Voyager Web Access

Generating a Certificate and Private Key

Installing a Certificate and Private Key

Password and Account Management

```
password-expiration <never | 1-1827>
expiration-warning-days <1-366>
expiration-lockout-days <never | 1-1827>
force-change-when <no | password | first-password>
deny-on-fail enable <on | off>
deny-on-fail failures-allowed <2-1000>
deny-on-fail allow-after <60-604800>
deny-on-nonuse enable <on | off>
deny-on-nonuse allowed-days <30-1827>
```

Users and Roles Management

Managing System Users

Managing Roles

Changing the Admin and Monitor Password

Configuring S/Key for Admin and Monitor

Show Commands

Group Management

Managing Groups

Show Commands

VPN Acceleration

Configuring VPN Acceleration

Displaying VPN Accelerator Information

Routing Commands

Route Map Commands

Set Routemap Commands

Show Routemap Commands

Routemap Protocol Commands

```
      off
      preference <1-65535> [family <inet | inet6 | inet-and-
      inet6>] on
```


Supported Route Map Statements by Protocol

Route Map Examples

```
      set routemap direct-to-ospf id 10 on
      set routemap direct-to-ospf id 10 match interface eth3c0
      set routemap direct-to-ospf id 10 match protocol direct
      set routemap direct-to-ospf id 10 action route-type type-2
      set routemap direct-to-ospf id 10 action metric value 20

      set ospf export-routemap direct-to-ospf preference 1 on
      set routemap rip-in id 10 on
      set routemap rip-in id 10 restrict
      set routemap rip-in id 10 match neighbor 10.1.2.3

      set routemap rip-in id 15 on
      set routemap rip-in id 15 match neighbor 10.1.2.4

      set routemap rip-in id 20 on
      set routemap rip-in id 20 action metric add 2

      set rip import-routemap rip-in preference 1 on
      set routemap static-to-bgp id 10 on
      set routemap static-to-bgp id 10 restrict
      set routemap static-to-bgp id 10 match protocol static
      set routemap static-to-bgp id 10 match network 10.0.0.0/8 all

      set routemap static-to-bgp id 15 on
      set routemap static-to-bgp id 15 match protocol static
      set routemap static-to-bgp id 15 action metric 100
      set routemap static-to-bgp id 15 action aspath-prepend-count
      4

      set routemap bgp-out id 10 on
      set routemap bgp-out id 10 match aspath-regex "(100 200+)"
```

```
origin any
set routemap bgp-out id 10 action metric 200
set bgp external remote-as 400 export-routemap bgp-out
preference 1 family inet on
set bgp external remote-as 400 export-routemap static-to-bgp
preference 2 family inet on
set routemap ospf3-to-bgp id 10 on
set routemap ospf3-to-bgp id 10 match protocol ospf3  (OSPF3
INTERNAL ROUTES)
set routemap ospf3-to-bgp id 10 action community replace on
set routemap ospf3-to-bgp id 10 action community no-export on
set routemap ospf3-to-bgp id 10 action community 200 as 100 on
set routemap ospf3-to-bgp id 10 action nexthop ipv6
3003::abcd:1012
```

..... 379

```
set routemap ospf3-to-bgp id 20 on
set routemap ospf3-to-bgp id 20 match protocol ospf3ase  (FOR
AS EXTERNAL ROUTES)
set routemap ospf3-to-bgp id 20 action community replace on
set routemap ospf3-to-bgp id 20 action community no-export on
set routemap ospf3-to-bgp id 20 action community 200 as 100 on
set routemap ospf3-to-bgp id 10 action nexthop ipv6
3003::abcd:1012

set routemap bgp-out id 10 on
set routemap bgp-out id 10 action community replace on
set routemap bgp-out id 10 action community none on
set routemap ospf3-to-bgp id 10 action nexthop ipv6
3003::abcd:1012

set bgp external remote-as export-routemap bgp-out preference
1 family inet6 on
set bgp external remote-as export-routemap ospf3-to-bgp
preference 2 family inet6 on
```

BGP

External BGP

BGP Peers

```
      log-warnings <on | off>
      trace bgp_traceoption <on | off>
      capability <default | ipv4-unicast | ipv6-unicast>
```

BGP Confederations

BGP Route Reflection

BGP Route Dampening

```
      keep-history <2-5400>
      keep-history default
```

Internal BGP

```
set bgp internal ..... 393
      <on | off>
      description text
      med <0-65535>
      med default
      outdelay <0-65535>
      outdelay off
      nexthop-self <on | off>
      local-address ip_address <on | off>
      virtual-address <on | off>
      interface [all | if_name] <on | off>
      protocol [all | bgp_internal_protocol] <on | off>
      peer ip_address peer_type <on | off>
      peer ip_address weight <0-65535>
      peer ip_address weight off
      peer ip_address no-aggregator id <on | off>
      peer ip_address holdtime <6-65535>
      peer ip_address holdtime default
      peer ip_address keepalive <2-21845>
      peer ip_address keepalive default
      peer ip_address ignore-first-ashop <on | off>
      peer ip_address send-keepalives <on | off>
      peer ip_address send-route-refresh [request | route-update]
      [ipv4 | ipv6 | All] [unicast]
      peer ip_address accept-routes all
      peer ip_address accept-routes none
      peer ip_address passive-tcp <on | off>
      peer ip_address authtype none
      peer ip_address authtype md5 secret secret
      peer ip_address throttle-count <0-65535>
      peer ip_address throttle count off
      peer ip_address log-state-transitions <on | off>
      peer ip_address log-warnings <on | off>
      peer ip_address trace bgp_traceoption <on | off>
      peer ip_address capability <default | ipv4-unicast | ipv6-
```

BGP Communities

BGP Show Commands

OSPF

OSPF Areas

OSPF Interfaces

OSPF Virtual Links

OSPF Global Settings

OSPF Show Commands

```
      interfaces detailed
      interface ifname
      interface ifname stats
      interface ifname detailed
      packets
      errors
      errors dd
      errors hello
      errors ip
      errors lsack
      errors lsr
      errors lsu
      errors protocol
      events
      border-routers
      database
      database areas
      database area ospf area
      database checksum
      database database-summary
      database detailed
      database external-lsa
      database inter-area-prefix
      database inter-area-router-lsa
      database intra-area-prefix-lsa
      database link-lsa
      database network-lsa
      database router-lsa
      database type <1-5>
      database events
      summary
```

RIP

RIP Interfaces

```
set rip interface if_name ..................................... 421
      off
      version <1 | 2> on
```

```
      metric <0-16>
      metric default
      accept-updates <on | off>
      send-updates <on | off>
      transport multicast
      transport broadcast
      authtype none
      authtype simple password
      authtype md5 secret secret [cisco-compatibility] <on | off>
      virtual address <on | off>
```

General RIP Properties

RIP Show Commands

IGRP

General IGRP Properties

```
      default-reliability <0-255>
      default-reliability off
      default-load <1-255>
      default-load off
      default-mtu <1-65535>
      default-mtu off
      k1 <0-16777215>
      k1 default
      k2 <0-16777215>
      k2 default
      holddown <on | off>
      max-hop-count <1-255>
      max-hop-count default
      update-interval <1-65535>
      update-interval default
      invalid-interval <1-65535>
      invalid-interval default
      hold-interval <1-65535>
      hold-interval default
      flush-interval <1-65535>
      flush-interval default
      validate fields <on | off>
```

IGRP Interfaces

```
set igrp interface if_name ..............................  427
      <on | off>
      delay <1-16777215>
      bandwidth <1-6777215>
      accept-updates <on | off>
```

IGRP Show Commands

```
show igrp.................................................  430
      errors
      interfaces
      interface if_address
      neighbors
      packets
      policy
      route-stats
```

IGMP

IGMP Commands

IGMP Show Commands

PIM

PIM Interfaces

PIM With IP Clustering

Sparse Mode PIM

```
      bootstrap-candidate priority default
      candidate-rp <on | off>
      candidate-rp local-address ip_address
      candidate-rp priority <0-255>
      candidate-rp priority default
      candidate-rp multicast group mcast_ip_prefix <on | off>
      static-rp off
      static-rp rp-address ip_iddresss < on | off>
      static-rp rp-address ip_address multicast-group mcast_ip_prefix
      <on | off>
      register-suppress-interval <60-3600>
      register-suppress-interval default
      candidate-rp advertise-interval <1-3600>
      candidate rp-advertise-interval default
      cisco compatibility <on | off>
      spt-threshold multicast mcast_ip_prefix threshold <0-1000000>
      <on | off>
      spt-threshold multicast mcast_ip_prefix threshold infinity
      <on | off>
```

Timer and Assert Rank Parameters for Dense Mode and Sparse Mode

```
set pim.................................................. 437
      hello-interval <1-21845>
      hello-interval default
      data-interval <11-3600>
      data-interval default
      assert-interval <1-3600>
      assert-interval default
      assert-limit <10-10000>
      assert-limit default
      assert-limit <0>
      jp-interval <1-3600>
      jp-interval default
      jp-delay-interval <1-3600>
      jp-delay-interval default
      jp-suppress-interval <2-3600>
      jp-suppress-interval default
      assert-rank protocol protocol name rank <0-255>
      assert-rank protocol protocol name rank default
```

Show PIM Commands

Route Aggregation

BOOTP

BOOTP Interfaces

BOOTP Show Commands

DVMRP

DVMRP Interfaces

DVMRP Timers

```
      cache-lifetime <60-86400>
      cache-lifetime default
```

DVMRP Show Commands

```
show dvmrp..................................................... 452
      interfaces
      interfaces if_name
      neighbors
      neighbor ip_address
      stats
      mfc
      reports
      route
      neighbor-routes
      summary
```

Static Routes

Configuring Static Routes

```
set static-route ip_prefix..................................... 453
      nexthop gateway address gateway_address priority <1-8> on
      nexthop gateway logical gateway_address priority <1-8> on
      nexthop gateway address gateway_address off
      nexthop gateway logical gateway_address off
      nexthop reject
      nexthop blackhole
      off
      rank default
      rank <0-255>
set static-route default....................................... 455
      next hop gateway address gateway_address priority <1-8> on
      nexthop gateway logical gateway_address priority <1-8> on
      nexthop gateway address gateway_address off
      nexthop gateway logical gateway_address off
      nexthop reject
      nexthop blackhole
      ip_prefix off
```

Static Multicast Routes

ICMP Router Discovery

ICMP Router Discovery Interfaces

ICMP Router Discovery Show Commands

IP Broadcast Helper

IP Broadcast Helper Forwarding

IP Broadcast Helper Interfaces

IP Broadcast Helper Show Commands

Network Time Protocol

Configuring an NTP Server

Adding an NTP Server

Deleting an NTP Server

NTP Show Commands

Dial on Demand Routing

Dial on Demand Routing Commands

Routing Option Commands

Equal-cost Path Splitting (Load Sharing)

Protocol Rank

Trace Routing Commands

Configuring the Trace Log File

Trace Option Variables

Show Route Summary Commands

Route Summary Commands

Show Routing Daemon (IPSRD) Commands

Show MFC Commands

Traffic Management Commands

Access Control List Commands

ACL Node Commands

ACL Ruleset Commands

Aggregation Class Commands

Set, Change, and View Aggregation Classes

Queue Class Commands

Set, Change, and View Queue Classes

ATM QoS

Configuring ATM QoS Descriptors

DSCP to VLAN Priority Commands

Configuring DSCP to VLAN Mapping

Monitoring Commands

Current and Historical Network Reports

Saving Reports to Files

Configuring How Much Data is Stored

Configuring CPU Utilization Reports

Configuring Memory Utilization Reports

Configuring Interface Linkstate Reports

Configuring Rate Shaping Bandwidth Reports

Configuring Interface Throughput Reports

Useful System Information

Displaying Useful System Statistics

Displaying Interface Settings

Displaying System Logs

Displaying Interface Traffic Statistics

Displaying the Interface Monitor

Displaying Resource Statistics

Displaying the Forwarding Table

Displaying Hardware Monitors

Displaying a Temperature Sensor Information

Command-Line Utilities

Index

C

D

E

F

G

H

I

O

P

Q

R

S

T

U

V

W

X

We Welcome Your Comments

Nokia Business Security Products is interested in improving our documentation to better serve you. Please feel free to send comments and suggestions to docfeedback@nokia.com.

If you are using Adobe Acrobat Reader 6.0 or later, we invite you to provide feedback to us by using the following form.

How satisfied are you with the help you received from this document? Choose One

Feel free to elaborate on your answer:

Where did you find this document? Choose One

If you chose “other,” where did you get the document?

Was the document easy to find? Choose One

Feel free to enter suggestions for improving this document:

May we contact you at your e-mail address if we have questions about your feedback? Yes

Submit

Note: This form is returned to us through your e-mail. We respect your privacy and will not use your e-mail address for any other purpose than communication about this form.