

# 4

## Managing the System

This chapter describes general tasks associated with managing the ERX system.

<b>Topic</b>	<b>Page</b>
Overview	4-2
Naming the System	4-2
Configuring Timing	4-3
Using the CLI	4-5
Managing vty Lines	4-7
Configuring the System Automatically	4-10
Saving the Current Configuration	4-10
Customizing the User Interface	4-13
Sending Messages	4-20
Managing Files	4-22
Transferring Files	4-27
Using the Telnet Client	4-39
Configuring DNS	4-39
Troubleshooting	4-43
Monitoring the System	4-48

## Overview

---

Managing the ERX system involves a variety of tasks. This chapter covers those tasks associated with the system in general rather than specific networking protocols. Each section in the chapter covers a different topic; where appropriate, a section contains an overview of the topic, configuration tasks, and information about monitoring the associated settings.

For additional management information, CLI commands, and procedures, refer to the following table.

Task	Reference
Find detailed information on commands described in this chapter.	<i>ERX Command Reference Guide</i>
Configure the system as an SNMP agent.	<i>Chapter 3, Configuring SNMP</i>
Set system passwords.	<i>Chapter 6, Passwords and Security</i>
Write CLI macros.	<i>Chapter 7, Writing CLI Macros</i>
Boot the system.	<i>Chapter 8, Booting the System</i>
Manage line modules and SRP modules.	<i>Chapter 5, Managing Line Modules and SRP Modules</i>

## Naming the System

---

When you receive the system, it has a factory default host name. To rename the system, use the **hostname** command.

### **hostname**

- Use to rename the system.
- The assigned name is displayed in the command line interface (CLI) prompts.
- Example

```
host1(config)#hostname host1
host1(config)#
```
- There is no **no** version.

## Configuring Timing

---

You can use the **timing source** command to configure three timing sources for the system. These sources are known as the primary, secondary, and tertiary sources. The system periodically polls the status of the current timing source. If the system discovers that the current source has become unavailable, it polls the timing source you specified as next in line. If this source is available, it switches to this source; if not, it then polls the next source in line. If the lowest source is unavailable, the system maintains the SRP clock as the source.

If you enable **auto-upgrade**, in the event of a source failure, the system—after switching to a lower source—polls all higher configured sources and automatically switches back to the highest timing source when that source becomes available.

The **timing select** command enables you to specify which source (primary, secondary, or tertiary) the system is to use by default. The system will never attempt to upgrade to a source higher than the selected source.

### *timing disable-auto-upgrade*

- Use to disable the **auto-upgrade** feature of the system's timing selector.
- The system starts out by setting the operational timing selector to the administratively configured selector. See the **timing select** command.
- Example  

```
host1(config)#timing disable-auto-upgrade
```
- The **no** version of this command restores the factory default, which is **auto-upgrade** enabled.

### *timing select*

- Use to specify which of the configured timing sources is used by default.
- Primary timing source is preferred over secondary, and secondary is preferred over tertiary. See the **timing source** command.
- If you enable the **auto-upgrade** feature, the system does not try to upgrade beyond the administratively configured selector.
- Example  

```
host1(config)#timing select secondary
```
- There is no **no** version.

### **timing source**

- Use to specify how the SRP module exchanges timing signals with an interface.
- You can specify primary, secondary, and tertiary timing sources.
- You can specify one external source received on an I/O module other than the SRP I/O module.
- You can specify two or more internal sources or external sources received via the SRP I/O module external timing ports.
- The available sources to choose are:
  - › ds1 – DS1 interface
  - › ds3 – DS3 interface
  - › e1 – E1 interface
  - › e3 – E3 interface
  - › sonet – SONET interface
  - › internal – internal system controller (SC) oscillator
  - › line – external timing input on SRP module
- Example

```
host1#timing source secondary sonet 3/0
```
- There is no **no** version.

### *Monitoring Timing*

Use the **show timing** command to view the timing settings for the system.

### **show timing**

- Use to display the timing settings and the operational status of the system timing.
- If a timing source fails, the system uses the next time source in the hierarchy, and a message appears in the system log at the *warning* level. If **auto-upgrade** is enabled, the system upgrades to a higher-priority timing source when one becomes available, and a message appears in the system log at the *notice* level.
- Example

```
host1#show timing
timing: tertiary (failover from primary)
primary: external SC E1 (A) (ERROR)
secondary: ds3 3/0 (ERROR)
tertiary: internal SC oscillator (ok)
auto-upgrade enabled
```

## Using the CLI

---

Use the commands described in this section to navigate the CLI. For a complete description of the CLI, see *Chapter 2, Command Line Interface*.

**configure**

- Use to enter Global Configuration mode.
- Global Configuration mode provides access to other configuration modes, such as Interface Configuration mode. See *Chapter 2, Command Line Interface*.
- This command allows other commands to be executed from a terminal or a file.

- Example 1

```
host1#configure
Configuring from terminal or file [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
host1(config)#
```

- Example 2

```
host1#configure
Configuring from terminal or file [terminal]? file
File name: system1.scr
Proceed with configure? [confirm]
host1(config)#
```

- There is no **no** version.

**disable**

- Use to exit Privileged Exec mode and return to User Exec mode.

- Example

```
host1#disable
host1>
```

- There is no **no** version.

### **enable**

- Use to move from User Exec to Privileged Exec mode.
- Privileged Exec mode allows you to access all other user interface modes. From here you can configure, monitor, and manage all aspects of the system.
- Set a password for this mode by using either the **enable password** or the **enable secret** command in Global Configuration mode. This protects the system from any unauthorized use.
- Once a password is set, anyone trying to use Privileged Exec mode will be asked to provide the password.

- Example

```
host1>enable
password:*****
host1#
```

- There is no **no** version.

### **end**

- Use to exit Global Configuration mode or any of the other Configuration modes. You may also use <Ctrl+Z> to exit these modes.
- Executing this command returns you to the User Exec mode.

- Example

```
host1(config)#end
host1#
```

- There is no **no** version.

### **exit**

- Use to exit the current command mode.

- Example

```
host1#exit
host1>
```

- There is no **no** version.

### **help**

- Use to display basic information about the interactive help system.
- Example

```
host1#help
```

Use the help options as follows:

```
?, or command<Space>? - Lists the set of all valid next keywords or arguments  
partial-keyword?      - Lists the keywords that begin with a certain character string  
partial-keyword<Tab> - Completes the partial keyword
```

- There is no **no** version.

### **run**

- Use to issue a User Exec mode from command from another CLI mode.
- Example

```
host1(config)#run show config | begin interface
```

- There is no **no** version.

### **sleep**

- Use to make the CLI pause for a specified period of time (in seconds).
- Pausing is very useful in configuration script files.
- Example

```
host1#sleep 60
```

- There is no **no** version.

## Managing vty Lines

---

The system supports 20 virtual tty (vty) lines for Telnet, SSH, and FTP services. Each Telnet, SSH, or FTP session requires one vty line. When you connect to the system via a vty line, the number of the vty line is not assigned sequentially; instead, the system assigns the first vty line that passes the host access list check rules.

### *Configuring vty Lines*

By default five vty lines (0–4) are open. You can open additional lines using the **line vty** command. Once lines are open, login is enabled by default. Before users can access the lines, you must configure a password, disable login using the **no login** command, or configure AAA authentication on the lines.

### **line vty**

- Use to open or configure vty lines.
- You can specify a single line or a range of lines. The range is 0–19.
- Example

```
host1(config)#line vty 6 10
host1(config-line)#
```
- Use the **no** version to remove a vty line or a range of lines from the configuration. Lines that you remove will no longer be available for use by Telnet, FTP, or SSH. When you remove a vty line, the system removes all lines above that line. For example, **no line vty 6** causes the system to remove lines 6 through 19. You cannot remove lines 0 through 4.

### **password**

- Use to specify a password on a single line or a range of lines.
- If you enable login but do not configure a password, the system will not allow you to access virtual terminals.
- Specify a password in plain text (unencrypted) or cipher text (encrypted). In either case, the system stores the password as encrypted.
- You can use the following keywords:
  - › **0** (zero) – specifies an unencrypted password
  - › **5** – specifies a secret
  - › **7** – specifies an encrypted password
- Example 1 (unencrypted password)

```
host1(config-line)#password 0 mypassword
```
- Example 2 (secret)

```
host1(config-line)#password 5 y13_x
```
- Example 3 (encrypted password)

```
host1(config-line)#password 7 x13_2
```
- Use the **no** version to remove the password. By default, **no password** is specified.

For more information about configuring security for vty lines, see *Chapter 6, Passwords and Security*.

### *Clearing vty Lines*

Use the **clear line** command to clear a vty line. Using this command terminates any service, such as an FTP session, on this line and closes any open files.

**clear line**

- Use to remove any services on a vty line and close any files opened as a result of services on that line.
- Specify the number of the vty line.
- Example

```
host1#clear line 2
```
- There is no **no** version.

*Monitoring vty Lines*

Use the **show line vty** command to monitor vty lines.

**show line vty**

- Use to display the configuration of a vty line.
- Field descriptions
  - › access-class – access-class associated with the vty line
  - › data-character-bits – number of bits per character
    - 7 – setting for the standard ASCII set
    - 8 – setting for the international character set
  - › exec-timeout – time interval that the terminal waits for expected user input
    - Never – indicates that there is no time limit
  - › exec-banner – status for the exec banner: enabled or disabled. This banner is displayed by the CLI after user authentication (if any) and before the first prompt of a CLI session.
  - › motd-banner – status for the MOTD banner: enabled or disabled. This banner is displayed by the CLI when a connection is initiated.
  - › login-timeout – time interval during which the user must log in.
    - Never – indicates that there is no time limit
- Example

```
host1#show line vty 0
no access-class in
data-character-bits 8
exec-timeout 3w 3d 7h 20m 0s
exec-banner enabled
motd-banner enabled
login-timeout 30 seconds
```

## Configuring the System Automatically

---

You can create an autoconfiguration script that runs whenever you reset the system. The following guidelines apply:

- You must name the script `autocfg.scr`.
- The script must begin with the following lines:

```
enable
conf t
```
- Add the commands desired to configure the system.
- For some configuration tasks, you might need to pause the CLI for 10 or so seconds by adding a **sleep** *seconds* command.



**Note:** The `autocfg.scr` script is bypassed if you arm the system to load from a script (not `autocfg.scr`) via the **boot config** or **boot backup** commands.

## Saving the Current Configuration

---

By default, the system automatically saves any change to the system configuration to nonvolatile storage (NVS). This feature is known as Automatic Commit mode, but has no effect on the CLI prompt. You can disable this feature by issuing the **service manual-commit** command. In Manual Commit mode (again with no effect on the CLI prompt), any configuration change affects only the current system configuration (the running configuration).

If you are in Manual Commit mode and want to save the configuration changes to NVS, you must issue either the **write memory** command or the **copy running-configuration startup-configuration** command.

If you change the configuration while in Manual Commit mode and issue the **reload** command without saving the changes to the startup configuration, the system provides a warning, allowing you to save the changes before reloading.

### **copy running-configuration**

- Use to save the current configuration to a system configuration (\*.cnf) file.
- This command is available only if the system is in Automatic Commit mode.
- The destination filename must have a .cnf extension.
- The destination file can be either a local or a network file.
- If you want to restore a previously saved configuration, use the **boot config filename** command.

- Example  

```
host1#copy running-configuration system2.cnf
```
- There is no **no** version.

### ***copy running-configuration startup-configuration***

- Use to save all outstanding (unsaved) configuration changes to NVS.
- This command is an exact alias of the **write memory** command.
- This command is available if the system is in either Automatic Commit mode or Manual Commit mode. If issued while in Automatic Commit mode, the CLI notifies you that the command is not necessary, but allows you to proceed.
- Example  

```
host1#copy running-configuration startup-configuration
```
- There is no **no** version.

### ***copy startup-configuration***

- Use to copy the previously saved startup configuration to a system configuration (\*.cnf) file. If you have made but not saved any configuration changes, those changes are not in the startup configuration.
- This command is available only if the system is in Manual Commit mode.
- Example  

```
host1#copy startup-configuration system1.cnf
```
- There is no **no** version.

### ***service manual-commit***

- Use to stop the system from automatically saving configuration changes to NVS.
- Issuing this command places the system into Manual Commit mode. This mode has no effect on the CLI prompt.
- Issuing this command causes an immediate save of configuration data not yet committed to NVS.
- Example  

```
host1(config)#service manual-commit
```
- Use the **no** version to revert to Automatic Commit mode; the **no** version has no effect if the system is already in Automatic Commit mode.

## **show configuration**

- Use to display the current (running) configuration of the system, a specified virtual router, or a specified interface within the current VR context.
- You can create a configuration script from the output by saving it as a file with the .scr extension.
- You can exclude information about particular types of interfaces.
- This command was formerly documented as **show config**; that abbreviation is still supported.
- You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. See *Chapter 2, Command Line Interface*, for details.
- This command is available only if the system is in Automatic Commit mode.
- Example

```

host1#show configuration
! Configuration script being generated on TUE JAN 29 200X
  00:31:12 UTC! Juniper Networks Edge Routing Switch ERX-700
! Version: x.y.z (January 18, 200X 15:01)
! Copyright (c) 1999-200X Juniper Networks, Inc. All rights
  reserved.
! Juniper Networks Edge Routing Switch ERX-700
boot config running-configuration
boot system erx_x-y-z.rel
no boot backup
no boot subsystem
no boot backup subsystem
no boot force-backup
no boot slot
!
! Note: The following commands are here to ensure that all
  virtual routers and
! vrfs are created before other commands that may need to
  reference them.
! These commands will be repeated further on as each virtual
  router and vrf
! has its configuration presented.
!
virtual-router default
virtual-router boston
!
ip vrf vpna
virtual-router vrA
!
hostname host1
exception protocol ftp anonymous null
!
controller t1 6/0
  channel-group 2 timeslots 1,3-8,10 speed 64

```

```
.  
. .  
. .  
!  
virtual-router vrA  
aaa authentication ppp default radius  
aaa accounting ppp default radius  
!  
ip address-pool local  
interface null 0  
ip bgp-community new-format  
no ip source-route  
!  
snmp-server  
!  
! End of generated configuration script.
```

### ***write memory***

- Use to save all outstanding (unsaved) configuration changes to NVS.
- This command is an exact alias of the **copy running-configuration startup-configuration** command.
- This command is available if the system is in either Automatic Commit mode or Manual Commit mode. If issued while in Automatic Commit mode, the CLI notifies you that the command is not necessary, but allows you to proceed.
- Example  

```
host1#write memory
```
- There is no **no** version.

### ***show running-configuration***

- Use to display the configuration currently running on the system.
- This command is available only if the system is in Manual Commit mode.
- Example  

```
host1#show running-configuration
```

## Customizing the User Interface

---

You can access the CLI via a console connected directly to the system or via a Telnet session. This section describes how you can customize the user interface. Some commands apply to the console, and some commands apply to vty lines that support Telnet sessions.

### *Setting the Console Speed*

You can specify the console speed for only the current console session or for the current console session and all subsequent console sessions.

#### ***speed***

- Use to set the speed for the current and all subsequent console sessions immediately.
- Example

```
host1(config)#line console 0
host1(config-line)#speed 14400
```
- Use the **no** version to revert to the default, 9600 bps.

#### ***terminal speed***

- Use to set the speed for the current console session.
- Example

```
host1#terminal speed 14400
```
- There is no **no** version.

### *Configuring the Display Terminal*

You can specify the number of lines that appear on a terminal screen and the number of characters that appear on a line.

#### ***terminal length***

- Use to set the number of lines on a screen.
- If a command generates more lines than the number configured, the output pauses after each screen.
- Set the number of lines on a screen in the range 0–512.
- Use 0 for no pausing.
- Example

```
host1#terminal length 25
```
- There is no **no** version.

#### ***terminal width***

- Use to set the width of the display terminal.
- Set the number of characters on a screen line in the range 30–512.
- Example

```
host1#terminal width 80
```
- There is no **no** version.

### *Specifying the Character Set*

You can specify the number of data bits per character for the current vty session and for all subsequent sessions on the specified vty lines. This feature allows you to display international characters on the terminal's screen.

#### ***data-character-bits***

- Use to set the number of bits per character on the terminal's screen for all future sessions on the specified lines.
- Use the default setting, 8, to view the full set of 8-bit international characters. Be sure that the software on other devices in the network also supports international characters.
- Set the number of bits to 7 to view only characters in the standard ASCII set.
- Example

```
host1(config)#line vty 1 3
host1(config-line)#data-character-bits 7
```
- There is no **no** version.

#### ***terminal data-character-bits***

- Use to set the number of bits per character on the terminal's screen for the current session.
- Use the default setting, 8, to view the full set of 8-bit international characters. Be sure that software on other devices in the network also supports international characters.
- Set the number of bits to 7 to view only characters in the standard ASCII set.
- Example

```
host1#terminal data-character-bits 7
```
- There is no **no** version.

### *Configuring Login Conditions*

You can issue the **dsr-detect** command to configure the system so that a data set ready (DSR) signal is required to log in to the console. If a session is in progress and the DSR signal is lost, the user is logged out automatically.

```
host1(config)#line console 0
host1(config-line)#dsr-detect
```

DSR is carried on pin 6 of the SRP module's RS-232 (DB-9) connector. The DSR input must be connected to the DSR output of a modem or the DTR output of another DTE device, such as a terminal server, that supports this signal.

### ***dsr-detect***

- Use to require that a DSR signal be detected on the line for a user to log in to the console.
- By default, DSR is not required and DSR detection is disabled.
- Example

```
host1(config-line)#dsr-detect
```
- Use the **no** version to remove the DSR requirement for login.

### *Setting Time Limits for User Login*

You can specify a time interval that the CLI waits for a user to provide a password when logging in to the console or a vty line. To do so:

- 1 Access the line configuration mode using either the **console** or **vtty** keyword.
- 2 Specify the time during which the user must enter the password. For example:

```
host1(config)#line console 0  
host1(config-line)#login  
host1(config-line)#boston  
host1(config-line)#timeout login response 15
```

### ***timeout login response***

- Use to set the time interval that the console or vty lines wait for the user to log in.
- If the interval passes and the user has not responded, the system closes the session or lines.
- Specify an interval in the range 0–300 seconds. A value of 0 means that there is no time limit during which the user must respond.
- The default value is 30 seconds.
- Example

```
host1(config-line)#timeout login response 15
```
- Use the **no** version to restore the default interval, 30 seconds.

### *Setting Time Limits for User Input*

You can specify a time interval that the CLI waits for user input on the console or vty lines. To do so:

- 1 Access the line configuration mode using either the **console** or **vtty** keyword.

- 2 Specify the time during which the user must enter information. For example:

```
host1(config)#line vty 0
host1(config-line)#exec-timeout 4192 13
```

### ***exec-timeout***

- Use to set the time interval that the console or vty lines wait for expected user input.
- If the interval passes and the user has not responded, the system closes the session or lines.
- Specify a time limit in the range 0–35791 minutes, and optionally specify the number of seconds.
- By default, there is no time limit.
- Example

```
host1(config-line)#exec-timeout 4192 13
```
- Use the **no** version to remove the time limit.

## *Configuring CLI Messages*

You can configure text banners for the CLI to display to users at different times in the connection process.

### ***banner***

- Use to configure message-of-the-day (MOTD), login, or exec banner to be displayed by the CLI:
  - › motd – displays the banner when a console or vty connection is initiated
  - › login – displays the banner before any user authentication (line or RADIUS authentication). The banner is also displayed if user authentication is not configured.
  - › exec – displays the banner after user authentication (if any) and before the first prompt of a CLI session
- If you do not specify an option, the default behavior is to display the banner as an MOTD.
- The first character in the banner string must be repeated at the end of the string; these characters delimit the banner. The CLI prompts you if you fail to repeat the opening delimiter. All text following the second occurrence of the delimiter is ignored without warning. The delimiter is case sensitive.
- Banner text can span multiple lines. It is truncated after 1,024 characters.
- Insert **\n** where you want the banner text to split and start a new line. Alternatively, you can press **<Enter>** on the CLI when you want the text to break. In the second case, you will be prompted for the remainder of the text after you press **<Enter>**. To display a backslash as part of the message, it must be immediately preceded by another backslash, like this: **\\**. Do not use a backslash as a delimiter or end a line with a backslash.

- To insert a ? character inside the text of a banner, you must enter <Ctrl+V> before entering the ? character. Failure to do so may produce undesired results.
- Examples

```

host1(config)#banner motd x This is an MOTD banner x
host1(config)#banner Y This is also an MOTD banner Y
host1(config)#banner "Quotes make good delimiters"
host1(config)#banner Xno space is required between the
delimiter and the real banner textX
host1(config)#banner b bad choice for a delimiter;
everything after that second b was ignored b
host1(config)#banner "This is one way\nto specify a
multiple line banner"
host1(config)#banner "This is another way to specify a
Enter remainder of text message. End with the character
' ' ' .
multiple line banner"

```

- Use the **no** version to remove the banner.

You can configure MOTD or exec banners, but not login banners, for the CLI to display on a per-line basis.

### ***exec-banner***

- Use to display an exec banner on a particular line after user authentication (if any) and before the first prompt of a CLI session.
- Banners on the lines are enabled by default; the **no** version does *not* reenables banners on the lines.
- See the **banner** command description for more information on configuring an exec banner.
- Example

```

host1(config-line)#exec-banner

```

- Use the **no** version to disable the exec banner on the line. If both the exec and MOTD banners are enabled on a line, issuing the **no exec-banner** command disables both the exec banner and the MOTD banner. The **no motd-banner** command behaves differently from the **no exec-banner** command.

### ***motd-banner***

- Use to display an MOTD banner on a particular line when a connection is initiated.
- Banners on the lines are enabled by default; the **no** version does *not* reenables banners on the lines.
- See the **banner** command description for more information on configuring an MOTD banner.
- Example

```

host1(config-line)#motd-banner

```

- Use the **no** version to disable the MOTD banner on the line. If both MOTD and exec banners are enabled on a line, issuing the **no motd-banner** command disables the MOTD banner and leaves the exec banner enabled. The **no motd-banner** command behaves differently from the **no exec-banner** command.

### *Monitoring the Console Settings*

You can use the following commands to monitor console settings.

#### ***show line console 0***

- Use to view the parameters configured for all future console sessions and the current console session.
- Example

```
host1#show line console 0
dsr-detect disabled
configured speed 9600, current speed 9600
exec-timeout never
```

#### ***show terminal***

- Use to view parameters of the current console session.
- Field descriptions
  - › Length – number of lines on the screen
  - › Width – number of characters on each line of the screen
  - › data-character-bits – number of bits per character
    - 7 – setting for the standard ASCII set
    - 8 – setting for the international character set
  - › Speed – speed of the console session
  - › dsr-detect – status of DSR signal detection
    - enabled – DSR signal must be detected for a user to log in to the console.
    - disabled – DSR signal need not be detected for a user to log in to the console.
  - › exec-timeout – time interval that the terminal waits for expected user input
    - Never – indicates that there is no time limit
  - › exec-banner – status for the exec banner: enabled or disabled. This banner is displayed by the CLI after user authentication (if any) and before the first prompt of a CLI session.
  - › motd-banner – status for the MOTD banner: enabled or disabled. This banner is displayed by the CLI when a connection is initiated.
  - › login-timeout – time interval during which the user must log in.
    - Never – indicates that there is no time limit

- Example

```
host1#show terminal
Length: 25 lines, Width: 80 columns
data-character-bits: 8 bits per character
Speed: 9600 bits per second
dsr-detect disabled
exec-timeout never
exec-banner enabled
motd-banner enabled
login-timeout 30 seconds
```

## Sending Messages

---

You can send a message to one or more terminals with the **send** command. You can specify a line number, a console number, or a vty number. You can also send the message to all terminals.

The following command sends the message “hello console” to line 0:

```
host1#send 0 "hello console"
```

The following command sends the message “hello everyone” to all terminals:

```
host1#send * "hello everyone"
```

If you begin the message on the same line as the **send** command, the first character of the message is considered to be a delimiter. You must use the same character to terminate the message. In both examples above, the delimiter was a double quotation mark (“”). If you press <Enter> without typing the second delimiter, the CLI prompts you for more message text and reminds you to complete the message with the delimiter, as shown in the following example:

```
host1#send vty4 XYou can start a message on the same line
Enter remainder of text message. End with the character
'X'.
and continue it on subsequent lines; the CLI prompts you for
Enter remainder of text message. End with the character
'X'.
more message text until you enter the second delimiterX
Proceed with send? [confirm]
```

If you do not begin the message on the same line as the **send** command, the CLI prompts you for the message text after you press <Enter>. The CLI does not recognize delimiters for these messages; you must enter <Ctrl+Z>, as shown in the following example:

```
host1#send 0
Enter remainder of text message. End with ^Z.
Good morning, Major Tom^Z
Proceed with send? [confirm]
```

The receiving terminals display the message without regard to other output currently displayed on the terminal. Pagination is not affected.

The sending terminal is not affected by the state of the intended receiving terminal. For example, if the receiving terminal is flow-controlled off or at a --More-- prompt, the message is still sent, and the sending terminal is available for further commands. The receiving terminal in this case displays the message when subsequently flow-controlled on or when the user responds to the --More-- prompt.

The receiving terminal displays the message, the line number of the sender, the username of the sender if the user was authenticated via RADIUS, and the time the message was sent.

### **send**

- Use to send a message to one or more terminals. You can specify a line number, a console number, or a vty number. You can use the \* keyword to send the message to all terminals.
- If you begin the message on the same line as the **send** command, the first character of the message is considered to be a delimiter. You must use the same character to terminate the message.
- The CLI prompts you for message text if you do not begin or complete the message on the same line as the **send** command. The CLI reminds you to signal the end of the message either with the delimiter or <Ctrl+Z>.
- Example

```
host1#send 0 "hello console"
```
- There is no **no** version.

## Managing Files

You are responsible for file management. Table 4-1 shows the types of system files and their corresponding extensions.

**Table 4-1** Types of system files and corresponding extensions

Type of File	Extension	Description
Configuration	*.cnf	Snapshot of the system's configuration
Core dump	*.dmp	File you can create for troubleshooting if a module fails
History	*.hty (reboot.hty)	Details of when and why modules rebooted
Log	*.log	A series of messages that describe events that occurred on the system
Macro	*.mac	A macro program
Release	*.rel	Software releases you can install in the system
Script	*.scr	A sequence of CLI commands. When you run a script file, the system executes the commands as though they were entered at the terminal
Secure Shell (SSH) Server public key	*.pub	Host key for the SSH server
Statistics	*.sts	Bulk statistics created when you run the <b>bulkstats</b> commands
Text	*.txt	Text file

System files may reside in four locations:

- The *system space*
- The *user space*
- A network host
- The standby SRP module

The system space contains files for system operation. For example, the current software configuration is stored in the system space.

The user space is reserved for FTP server operations and has the typical directory structure of a secure FTP server. The root or top level directory is a read-only directory that contains two subdirectories:

- */incoming* – read-write directory to and from which an FTP client can send and retrieve files.
- */outgoing* – read-only directory from which an FTP client can retrieve files.

Users can transfer files via FTP to the user space from a network host and vice versa. However, users cannot access the system space via FTP. To install a file from the user space to the system space, use the **copy** command. For detailed information on transferring files between locations, see *Transferring Files* later in this chapter.

In order to conserve NVS and minimize the installation time, files are not stored in both the system space and the user space. When you issue the **copy** command to install a file from user space to system space, the ERX system establishes a link to the file, but does not make a physical copy.

### *Managing the User Space from a Network Host*

If you enable the system's FTP server (see *Configuring the FTP Server* later in this chapter), you can manage files on the user space from an FTP client on a network host. Table 4-2 lists the FTP protocol commands that the ERX system supports. Whether you can perform these functions on the user space depends on the features that the FTP client offers.

**Table 4-2** FTP protocol commands that the system supports

FTP Protocol Command	Function
HELP	List supported commands.
USER	Verify user name.
PASS	Verify password for the user.
QUIT	Quit the session.
LIST	List contents of a directory.
NLST	List directory contents using a concise format.
RETR	Retrieve a file.
STOR	Store a file.
CWD	Change working directory.
CDUP	Change working directory to parent.
TYPE	Change the data representation type.
PORT	Change the port number.
PWD, XPWD	Get the name of current working directory.
STRU	Change file structure settings (only stream mode supported).
MODE	Change file transfer mode (only stream mode supported).
PASV	Make the server listen on a port for data connection.
NOOP	Do nothing.
DELE	Delete a file.
MKD, XMKD	Make directory.

**Table 4-2** FTP protocol commands that the system supports (continued)

FTP Protocol Command	Function
RMD, XRMD	Remove directory.
RNFR	Rename from (i.e., “from half” of file or directory rename)
RNTO	Rename to (i.e., “to half” of file or directory rename)

*File Commands and FTP Servers*

Commands—**copy**, **configure file**, and **macro**—that invoke a remote FTP server take place in the context of the current virtual router rather than the default VR. You must configure the remote FTP server so that any traffic destined for the virtual router can reach the virtual router; typically, you configure the FTP server to reach the default address of the system, which will always be able to reach the virtual router.

*Renaming Files*

To rename files, use the **rename** command. Table 4-3 shows the types of files you can rename in different locations.

**rename**

- Use to rename a local file.
- You can change the base name but not the extension of a file.
- Example
 

```
host1#rename boston1.cnf boston2.cnf
```
- There is no **no** version.

**Table 4-3** File types you can rename

Source	Destination			
	System Space	User Space (linked files and unlinked files)	Network Host Within a Firewall	Standby SRP Module
System	*.cnf	*.cnf	*.sts	None
	*.dmp	*.dmp		
	*.hty	*.hty		
	*.log	*.log		
	*.mac	*.mac		
	*.rel	*.scr		
	*.scr	*.txt		
	*.txt			
	Nonsystem files			

**Table 4-3** File types you can rename (continued)

Source	Destination			
	System Space	User Space (linked files and unlinked files)	Network Host Within a Firewall	Standby SRP Module
User Space	*.cnf *.hty (excluding reboot.hty) *.log (excluding system.log) *.mac *.scr *.txt	*.cnf *.dmp *.hty *.log *.mac *.pub *.rel *.scr *.sts *.txt Nonsystem files	None	None
Network Host Within a Firewall	None	None	None	None
Standby SRP Module	None	None	None	None

### Deleting Files

Use the **delete** command to delete files in NVS. Table 4-4 shows the types of files you can delete in different locations.

#### **delete**

- Use to delete files in NVS.
- To delete a file in user space, specify the incoming or outgoing directory on the FTP server.
- You can specify the name of a subdirectory in the incoming or outgoing directory.
- Examples:

```
host1#delete test.scr
host1#delete /outgoing/test.scr
```

- There is no **no** version.

**Table 4-4** File types you can delete

Location			
System Space	User Space (linked files and unlinked files)	Network Host <b>Within</b> a Firewall	Standby SRP Module
*.cnf	*.cnf	None	None
*.dmp	*.dmp		
*.hty	*.hty		
*.log	*.log		
*.mac	*.mac		
*.rel	*.pub		
*.scr	*.rel		
*.sts	(deletes *.rel file only and not associated files)		
*.txt	*.scr		
	*.sts		
	*.txt		
	Nonsystem files		

### Monitoring Files

Use the **dir** command to view files in NVS.

#### **dir**

- Use to show a list of files in NVS.
- Specify a directory path to view files in the user space.



**Note:** If you issue the **dir** command from Boot mode, existing *.scr* and *.mac* files are not displayed.

- Field descriptions
  - › file – name of file or directory (DIR indicates a directory)
  - › size – physical size of file
  - › unshared size – size of file in user space
    - value of zero indicates that this file has been installed onto the system space and that there is a link to this file
    - value other than zero indicates that the file has not been installed onto the system space and equals the physical size of the file
  - › date – date that file was created
  - › in use – an exclamation point (!) indicates that the system is using this file

- Examples

```
host1#dir
          file              size      unshared      size      date (UTC)      in
          -----          -
          /incoming <DIR>  38023824          0              0      12/19/2000 07:13:00  use
          /outgoing <DIR>   3584              0              0      12/19/2000 07:13:00  use
          reboot.hty       5632             5632           5632     12/20/2000 10:01:40  use
          3-0-0a3-7.rel    38797998         38797998       38797998 12/20/2000 23:40:46  !
          test.scr         1204             1204           1204     12/18/2000 03:01:04  use
```

Capacity = 220200960, Bytes Free = 120616448, Reserved = 36700160

```
host1#dir /incoming
          file              size      unshared      size      date (UTC)      in
          -----          -
          3-0-0a3-7.rel    256              0              0      12/19/2000 07:14:01  use
          srp.exe         30012312          0              0      12/19/2000 07:14:12  use
          srpIc.exe       1801208           0              0      12/19/2000 07:20:32  use
          srpDiag.exe     6984222           0              0      12/19/2000 07:22:08  use
```

Capacity = 220200960, Bytes Free = 120616448, Reserved = 36700160

```
host1#dir /outgoing
          file              size      unshared      size      date (UTC)      in
          -----          -
          test.scr        1204              0              0      12/18/2000 03:01:04  use
          foo.scr         1278             1278           1278     12/20/2000 04:02:12  use
```

Capacity = 220200960, Bytes Free = 120616448, Reserved = 36700160

- There is no **no** version.

## Transferring Files

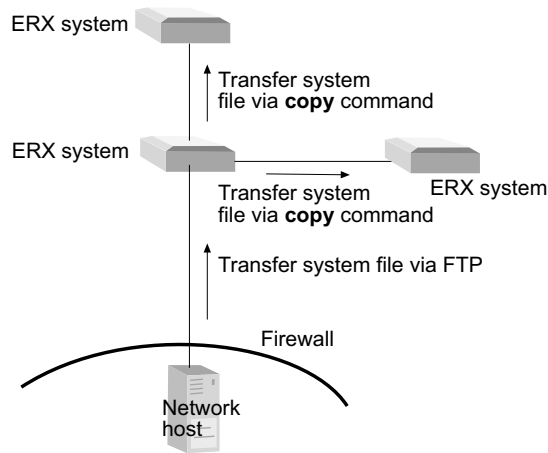
You may need to transfer files between the following locations:

- System space
- User space
- Network host
- Standby SRP module

There are two ways of transferring files: using the **copy** command and using the system's FTP server. Table 4-5 shows the types of files that you can transfer between the locations using the **copy** command, which activates a hidden FTP client on the ERX system.

The system's FTP server allows the transfer of files between a network host and the user space. When a firewall separates the ERX system from the network host, you must use the FTP server to transfer files to the user space. You can then install the files from the user space to the system space using the **copy** command. However, if there is no firewall between the ERX system and the network host, you can use the **copy** command or the FTP server to transfer files.

For example, you can transfer a file from a network host to an ERX system via FTP, and then transfer the file via the **copy** command from the ERX system to other ERX systems. See Figure 4-1.



**Figure 4-1** Transferring system files to the ERX system

### *Using the copy Command*

Table 4-5 shows the types of files that you can transfer between the locations by using the **copy** command.

**Table 4-5** File types you can transfer using the **copy** command

Source	Destination			
	System	User Space (linked files and unlinked files)	Network Host Within a Firewall	Standby SRP Module
System	*.cnf *.hty (excluding reboot.hty) *.log (excluding system.log) *.mac *.scr *.txt	*.cnf *.hty *.log *.mac *.pub *.scr *.txt	*.cnf *.dmp *.hty *.log *.mac *.pub *.scr *.sts *.txt	None
User Space	*.cnf *.mac *.rel *.scr *.txt	*.cnf *.hty *.log *.mac *.pub *.rel (*rel file only, not files associated with the *.rel file) *.scr *.txt Nonsystem files	None	None
Network Host Within a Firewall	*.cnf *.mac *.rel *.scr *.txt	None	None	None
Standby SRP Module	system.log reboot.hty	system.log reboot.hty *.dmp	system.log reboot.hty *.dmp	None

To transfer files via the **copy** command between the system space and a network host:

- 1 Check whether there is a route to the network host, and create one if necessary. See *ERX Routing Protocols Configuration Guide, Vol. 1, Chapter 2, Configuring IP*.
- 2 Configure the network host as an FTP server.



**Note:** This command takes place in the context of the current virtual router (VR) rather than the default VR. You must configure the FTP server so that any traffic destined for the VR can reach the VR; typically, you configure the FTP server to reach the default address of the ERX system, which will always be able to reach the VR.

- 3 Add the FTP server to the static host table, so that the ERX system can access the network host.
- 4 (Optional) Specify a source interface to use in FTP packets leaving the router.
- 5 Copy the files.

### **copy**



- Use to copy a file from one location to another.

**Note:** You cannot copy script (.scr) or macro (.mac) files while in Boot mode. You can copy only .cnf, .hty, and .rel files. If you issue the **dir** command from Boot mode, existing .scr and .mac files are not displayed.

- See Table 4-1 for the types of files you can copy.
- Specify a network path to copy to or from another device on the network.
- Specify the incoming or outgoing directory to copy to or from the user space.
- Specify a subdirectory name to create a subdirectory within the incoming or outgoing directory in the user space.
- You cannot use wildcards.
- You cannot create or copy over files generated by the system; however, you can copy such files to an unreserved filename.
- Examples

```
host1#copy host1:westford.cnf boston.cnf
host1#copy /incoming/releases/2-8-0a3-7.rel 2-8-0a3-7.rel
```

- There is no **no** version.

### **host**

- Use to add or modify an entry to the host table.
- Specify the number 8 before the user name and before the password to encrypt these values. By default, the user name and password are not encrypted.
- This command allows network files to be accessible from a host.
- Example

```
host1(config)#host westford 10.10.8.7 ftp 8 user25 8
kxu83m41
```

- Use the **no** version to remove a specified host.

### *ip ftp source-address*

- Use to specify an operational interface by IP address as the source interface for FTP packets sent via the system's FTP client.
- This command overrides a setting you configured previously with the **ip ftp source-interface** command.
- If you issue this command, the output of the **show configuration** command includes an entry of the following format:

```
ip ftp source-address ipAddress
```

› *ipAddress* – IP address of the interface

This entry also appears in the output if you delete an interface or change its IP address after issuing the **ip ftp source-interface** command, in which case the IP address is the one that was configured on the interface before you issued the **ip ftp source-interface** command

- Example

```
host1(config)#ip ftp source-address 10.10.5.21
```

- Use the **no** version to restore the default, in which the source address in the FTP packets is that of the interface where the FTP connection is made.

### *ip ftp source-interface*

- Use to specify an operational interface by interface type and location as the source interface for FTP packets sent via the system's FTP client.
- The interface you specify must have an IP address.
- This command overrides a setting you configured previously with the **ip ftp source-address** command.
- If you issue this command and the interface is valid, the output of the **show configuration** command includes an entry of the following format:

```
ip ftp source-interface interfaceType interfaceSpecifier
```

› *interfaceType* – type of interface

› *interfaceSpecifier* – location of the interface

For information about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.

- If you delete the interface or change its IP address, the output of the **show configuration** command appears as if you had entered the **ip ftp source-address** command:

```
ip ftp source-address ipAddress
```

› *ipAddress* – IP address of the interface when you issued the **ip ftp source-interface** command

- Example

```
host1(config)#ip ftp source-interface loopback1
```

- Use the **no** version to restore the default, in which the source address in the FTP packets is that of the interface where the FTP connection is made.

### *Configuring the FTP Server*

To transfer files via the system's FTP server, you must configure the FTP server and ensure that FTP client software is installed on the network host.

Although you can transfer any type of file via FTP to the ERX system, the principal aim of this feature is to allow the transfer of system files to NVS. You can transfer files via FTP to the user space. You can then install files from the user space onto the system using the **copy** command. It is not possible to access the system files directly via FTP operations.

FTP sessions on the ERX system use the vty lines. The ERX system divides its vty resources between Telnet, SSH, and FTP services. Each FTP session requires one vty line. The FTP service uses the authentication method configured for the vty lines.

#### Features

The system supports the following FTP features:

- Compliance with RFC 959 – File Transfer Protocol (FTP) (October 1985)
- FTP passive mode
- Efficient NVS organization
- User authentication via RADIUS or password checking

#### FTP Passive Mode

Normally, when a client connects to an FTP server, the client establishes the control channel with the server, and the server responds by opening a data channel to the client. However, when the FTP client and server are on opposite sides of a firewall that prohibits inbound FTP connections, the server cannot open a data channel to the client.

FTP passive mode overcomes this connection limitation. In passive mode, the client opens a control channel to the server, tells the server it wants to operate in passive mode, and opens the data channel to the server. This method of establishing the FTP connection allows both the control channel and the data channel to pass through the firewall in the allowed direction.

### Configuring Authentication

Before you enable the FTP server, configure the authentication procedure for the vty lines, as follows:

- 1 Configure host access lists.
- 2 Configure user authentication methods.
- 3 Configure the vty lines to use the host access lists and user authentication methods.

You can specify authentication via a RADIUS server or via password checking. If you choose no authentication service, any client can access the FTP server. For information about authentication on vty lines, see *Chapter 6, Passwords and Security*.

### Configuration Tasks

FTP is disabled by default. You must enable the FTP server with the **ftp-server enable** command before the system allows FTP clients to connect.

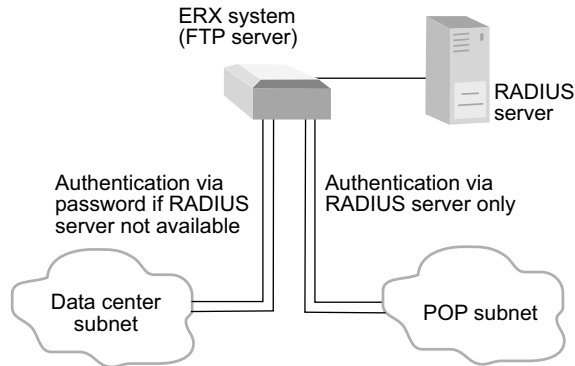
#### ***ftp-server enable***

- Use to enable the FTP server and to monitor the FTP port for attempts to connect to the FTP server.
- You can enable the FTP server on the default virtual router only.
- Example  

```
host1(config)#ftp-server enable
```
- Use the **no** version to terminate current FTP sessions and to disable the FTP server.

### Configuration Example

Figure 4-2 shows the scenario for this configuration example.



**Figure 4-2** FTP configuration example

In this example, two FTP lines are required for administrators on the data center subnet, and two more lines are required for users on the POP subnet. The system verifies passwords of administrators on the data center subnet via either a RADIUS server or via simple line authentication if the RADIUS server is unreachable. However, the system verifies passwords of users on the POP subnet only via the RADIUS server.

The following example shows all steps for configuring this scenario, from specifying a RADIUS server to enabling the FTP line:

- 1 Configure the RADIUS server.

```
host1(config)#radius authentication server 10.6.131.51
host1(config-radius)#key abc123
host1(config-radius)#udp-port 1645
```

- 2 Configure two access lists; one named “DataCenter,” permitting only the data center subnet, and one named “Pops,” permitting only the POP subnet.

```
host1(config)#access-list DataCenter permit 10.6.128.0
255.255.128.0
host1(config)#access-list DataCenter deny any
host1(config)#access-list Pops permit 199.125.128.0
255.255.128.0
host1(config)#access-list Pops deny any
```

- 3 Configure two authentication method lists, named “RadiusAndLine” and “RadiusOnly.”

```
host1(config)#aaa new-model
host1(config)#aaa authentication login RadiusAndLine radius
line
host1(config)#aaa authentication login RadiusOnly radius
```

- 4 Configure two FTP lines to be used by data center administrators.  

```
host1(config)#line vty 0 1
host1(config-line)#password foobar
host1(config-line)#access-class DataCenter in
host1(config-line)#login authentication RadiusAndLine
```
- 5 Configure the remaining FTP lines to be used by POP administrators.  

```
host1(config)#line vty 2 4
host1(config-line)#password foobar
host1(config-line)#access-class Pops in
host1(config-line)#login authentication RadiusOnly
```
- 6 Enable the FTP server.  

```
host1(config)#ftp-server enable
```

### *Monitoring the FTP Server*

Use the **dir** command to monitor files on the FTP server. Use the **show ftp-server** and **show users** commands to monitor settings of the FTP server.

#### ***show ftp-server***

- Use to display information about the FTP server.
- Field descriptions
  - › FTP Server state – status of the FTP server: enabled or disabled
  - › Open connections – number of open connections to the FTP server
  - › Statistics since server was last started – data about the connection attempts since you enabled the FTP server
  - › Statistics since last system reload – data about the connection attempts since you last booted the system
    - attempts – number of attempts to connect
    - failed hosts – number of connection attempts that failed because of disallowed host addresses
    - failed users – number of connection attempts that failed because users were not authenticated

- Example

```

host1#show ftp-server
FTP Server state: enabled, 0 open connections
Statistics since server was last started:
    attempts: 32
    failed hosts: 5
    failed users: 7
Statistics since last system reload:
    attempts: 35
    failed hosts: 5
    failed users: 8
    
```

### **show users**

- Use to display information about users of the vty lines.
- Specify the **detail** keyword to view detailed information.
- Field descriptions
  - › line number – number of the line to which the user is connected
  - › line name – name of the line and the service the line offers
  - › user – name of the user
  - › connected from – location or IP address of the user
  - › connected since – date and time that the user connected to the line
- Example

```

host1#show users
line          connected
number  line name  user    from    connected since
-----  -
0*      console 0           console 02/12/2001 19:57
4       vty 3 (ftp)  fred    10.10.0.64 02/12/2001 20:04
5       vty 4 (telnet) 10.10.0.64 02/12/2001 20:04
    
```

Note: '\*' indicates current user.

### *Copying Partial Releases*

You can shorten the time it takes to copy a release from a server and reduce the amount of storage needed for a release. At the default setting, all subsystems are included when you copy a release from a server. Use the **exclude-subsystem** command to specify subsystems that you do not want to copy from the server. Use the **show subsystems** command to verify which files are included and excluded when you copy a release from a server. Follow this example:

- 1 Determine which subsystems are included in the release on the server.

```

host1#show subsystems file m:/x/images/x-y-z.rel
    
```

- 2 Exclude any subsystems in the release that you do not need for the configuration.

```
host1#(config)#exclude-subsystem ct1
host1#(config)#exclude-subsystem coc12
host1#(config)#exclude-subsystem oc12s
```

- 3 (Optional) Remove a subsystem from the exclude list.

```
host1#(config)#no exclude-subsystem oc12s
```

- 4 (Optional) Verify the subsystems that will be included and excluded in future release copies.

```
host1#show subsystems file x8.rel
```

- 5 (Optional) After copying a release, view which subsystems were excluded.

```
host1#show configuration
...
exclude-subsystem ct1
exclude-subsystem coc12
```

- 6 (Optional) Determine whether the currently running software is a result of a copy with excluded subsystems. The word “Partial” indicates that subsystems were excluded.

```
host1#show version
Juniper Networks, Inc. Operating System Software
Copyright (c) 200X Juniper Networks, Inc. All rights
reserved.
System Release: x-y-z.rel Partial
```

### ***exclude-subsystem***

- Use to exclude any subsystems that are in a release that you do not need for the system configuration.
- Example
 

```
host1(config)#exclude-subsystem ct1
```
- The subsystems that you indicate are added to the “exclude list.” All subsequent release copies will exclude the images for these subsystems from the release copy.
- Example
 

```
host1(config)#no exclude-subsystem ct1
```
- Use the **no** version of this command *with the subsystem name* to remove a subsystem from the exclude list. Use the **no** version of this command *without a subsystem name* to remove *all* subsystems from the exclude list.

### ***show subsystems***

- Use to determine which subsystems are included in the current software release on the system or in a specified software release file.
- Specify either a local filename or remote path and filename to view the subsystems that are included in a software release file other than the current software release on the system.
- Field descriptions
  - › Required – number of bytes of data for the required portion of the release
  - › Included Subsystems – number of bytes of data for the included subsystems listed. All included subsystems in the release are listed.
  - › Excluded Subsystems – number of bytes of data for the excluded subsystems listed. All excluded subsystems in the release are listed.
- Use the command before you copy a release to verify which subsystems are present in the release.
- Example

```
host1#show subsystems file m:/x/images/x-y-z.rel
oc3
ct3
ut3f
ut3a
ct1
dpfe
oc12p
oc12a
ge
fe8
coc12
oc12s
```

- Use the command after copying a release to verify which subsystems are included and excluded.
- Example

```
host1#show subsystems file x8.rel
Required: 1423005 bytes
Included Subsystems: 27882192 bytes
ct3
ut3f
ut3a
dpfe
oc12p
oc12a
ge
fe8
coc12
oc12s
```

```
Excluded Subsystems: 6840211 bytes
oc3
ctl
```

## Using the Telnet Client

---

The system has an embedded Telnet client that enables you to connect to remote systems. You can configure a Telnet daemon to listen in virtual routers other than the default virtual router. You must be in the context of the desired virtual router to issue the command.

### ***telnet***

- Use to open a Telnet connection to a remote system.
- Specify the IP address or name of the remote host.
- You can specify a VRF context in which the request takes place.
- Depending on how the remote system accepts Telnet requests, you can specify a port number or port name through which the system will connect to the remote host. In the Transmission Control Protocol (TCP), ports define the ends of logical connections that carry communications. In most cases, you can accept the default, port number 23, the Telnet port. For more information on port numbers and associated processes, see [www.iana.org](http://www.iana.org).
- You can force Telnet to use the IP address of an interface that you specify as its source address.
- Example

```
host1#telnet 192.168.35.13 fastEthernet 0
```
- There is no **no** version.

### ***telnet listen***

- Use to create a Telnet daemon to listen in a virtual router.
- Example

```
host1(config)#virtual-router 3
host1:3(config)#telnet listen port 3223
```
- Use the **no** version of the command to delete the daemon.

## Configuring DNS

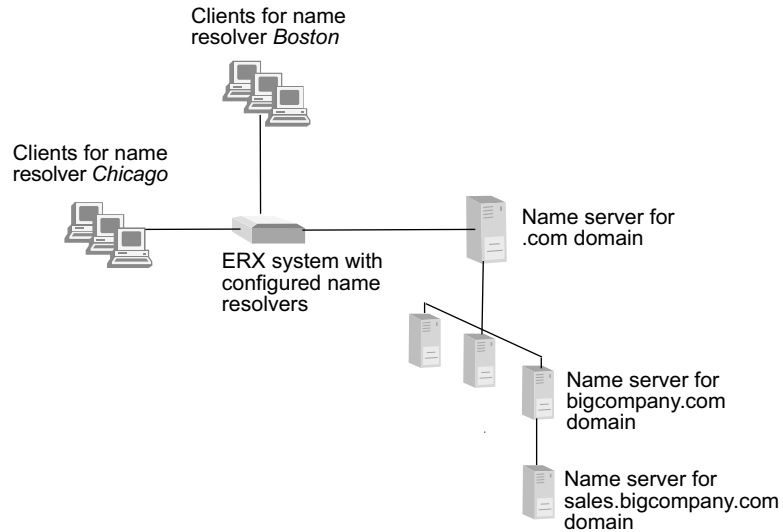
---

You can configure virtual routers to act as *name resolvers* for Domain Name Service (DNS). DNS is a client/server mechanism that maps IP addresses to hostnames.

The name resolver is the client side of DNS and receives address-to-hostname requests from its own clients when they want to contact hosts on other networks. By polling *name servers*, the name

resolver learns name-to-address translations for the hosts its clients want to contact.

A name server may provide the translation from its cache or may poll servers lower in the DNS hierarchy to obtain a translation. Typically, name servers at the top of the hierarchy recognize top level domain names and know which servers to contact for information about more detailed domain names. See Figure 4-3.



**Figure 4-3** DNS hierarchy example

DNS messages from a name resolver to a name server must include the domain name for the resolver's clients. Consequently, you must specify a default domain name for the clients. The default domain name is appended to unqualified hostnames (those without domain names).

The name resolver must be able to access at least one name server. Accordingly, you must configure a static route to a gateway that provides access to the name server and assign the name server to the name resolver. See *Assigning Name Servers*, later in this chapter.

Each virtual router can have its own name resolver and domain name. However, if two virtual routers use the same name servers and belong to the same local domain, you do not need to configure name resolvers on both virtual routers. See *Using One Name Resolver for Multiple Virtual Routers*, later in this chapter.

## References

For more information about the DNS, consult the following resources:

- *RFC 1035 – Domain Names – Implementation and Specification (November 1987)*
- *RFC 2308 – Negative Caching of DNS Queries (DNS NCACHE) (March 1998)*

## Assigning Name Servers

To assign name servers to the system:

- 1 Access the virtual router context.
- 2 Define static routes to the gateways that provide access to the name servers.
- 3 Enable the virtual router to query name servers.
- 4 Specify a default domain name for the hosts.
- 5 Specify the name servers.

### Example

```
host1(config)#virtual-router boston
host1:boston(config)#ip route 0.0.0.0 0.0.0.0
gatewayIpAddress
host1:boston(config)#ip domain-lookup
host1:boston(config)#ip domain-name urlofinterest.com
host1:boston(config)#ip name-server 10.2.0.3
host1:boston(config)#ip name-server 10.2.5.5
```

### *ip domain-lookup*

- Use to enable the system to query the configured DNS name servers when it needs an IP-hostname-to-IP-address translation.
- Domain lookup is disabled by default.
- Example

```
host1(config)#ip domain-lookup
```
- Use the **no** version to disable domain lookup.

### *ip domain-name*

- Use to define a default domain name for the clients that a name resolver serves.
- You must define a default domain name for each name resolver. Multiple name resolvers can use the same default domain name.
- If you map an unqualified hostname (one without a domain name) to an IP address with the **host ftp** command, the domain name is appended to the hostname before the name is stored in the host table.
- Example

```
host1(config)#ip domain-name bigcompany.com
```
- Use the **no** version to delete the domain name; that is, the domain name will no longer be appended to hostnames in the static host table.

### *ip name-server*

- Use to specify a DNS name server that the system can query for hostname-to-IP-address resolution.
- Example

```
host1(config)#ip name-server 192.168.25.100
```
- Use the **no** version to delete the name server.

### *Using One Name Resolver for Multiple Virtual Routers*

You can use one name resolver for multiple virtual routers if those virtual routers use the same name servers and belong to the same local domain. To do so, complete the following steps:

- 1 Configure a name resolver for the first virtual router.
- 2 Access the context for the second virtual router.
- 3 Specify that the second virtual router should use the name resolver you configured for the first virtual router.
- 4 Repeat steps 2 and 3 for other virtual routers that you want to point to this name resolver.

**Example** To configure the virtual router *boston* to use the same name servers as the default router, enter the following commands.

```
host1(config)#virtual router boston
host1:boston(config)#ip domain-lookup transit-virtual-router
default
```

### ***ip domain-lookup transit-virtual-router***

- Use to configure a virtual router to use the name servers you configured for another virtual router.
- Example
 

```
host1:boston(config)#ip domain-lookup transit-virtual-router
                        default
```
- Use the **no** version to stop a virtual router from using the same name servers you configured for another virtual router.

### *Monitoring DNS*

After you configure DNS, you can use the **show ip domain-lookup** command to view information about the name servers.

### ***show ip domain-lookup***

- Use to display the name servers that you have specified on the system with the **ip name-server** command.
- Field descriptions
  - › Bind to client – name of the virtual router context in parentheses, followed by the name of the virtual router providing the name resolver
  - › Using following Domain Name Servers – name servers you assigned
  - › Using following Local Domain Names – default domain names you specified
- Example
  - › In this example, the virtual router *boston* uses the name resolver on the default virtual router.
 

```
host1#show ip domain-lookup
Bind to client: (boston)default
Using following Domain Name Servers:
10.2.0.3
11.1.1.1
10.1.1.1
Using following Local Domain Names :
urlofinterest.com
concord
```
- Use the **no** version to disassociate this NFS server from the current virtual router.

## Troubleshooting

---

You can use **log** commands to discover and isolate problems with the system. For information on using the log commands, see *Chapter 11, Logging System Events*. You can also use dump files to troubleshoot line module failures.

## Creating Core Dump Files

You can enable the system to create a core dump file if a module fails.

You can choose to send the core dump file to an FTP server or save the file in a compressed form to NVS. Juniper Networks Customer Service can then access the core dump file and analyze it to determine what went wrong. The core dump is disabled by default. You can enable the core dump from Boot mode or Global Configuration mode.



**Caution:** Create a core dump file only under the direction of Juniper Networks Customer Service. Network function can be disrupted if you create a core dump file while the system is running in a network.

### Boot Mode

To enable the core dump from Boot mode:

- 1 Access Boot mode by reloading the SRP module; then press the <mb> key sequence (case insensitive) during the countdown.
- 2 Specify where the system should transfer the core dump file.
- 3 Set the IP address and mask of the system interface over which you want to send the core dump file.
- 4 Specify the gateway through which the system sends the core dump file to the FTP server.
- 5 (Optional) Set a username and password for FTP access to the server where you transferred the core dump file.
- 6 Reload the operating system.

### Example

```
:boot##exception dump 192.168.56.7 CORE_DUMPS
:boot##exception protocol ftp user_name user_password
:boot##exception gateway 192.168.12.3
:boot##exception source 10.10.33.8 255.255.255.0
:boot##reload
```

### Global Configuration Mode

To enable the core dump from Global Configuration mode:

- 1 Access Global Configuration mode.
- 2 Specify where the system should transfer the core dump file.
- 3 Set the IP address and mask of the system interface over which you want to send the core dump file.

- 4 Specify the gateway through which the system sends the core dump file to the FTP server.
- 5 (Optional) Set a username and password for FTP access to the server where you want to transfer the core dump file.
- 6 (Optional) View parameters associated with creating a core dump file.

**Example**

```
host1(config)#exception dump 192.168.56.7 CORE_DUMPS
host1(config)#exception protocol ftp username userpassword
host1(config)#exception gateway 192.168.12.3
host1(config)#exception source 10.10.33.8 255.255.255.0
host1(config)#reload
```

**exception dump**

- Use to specify where the system should transfer the core dump file.
  - › To send the file to an FTP server, enter the IP address of the FTP server and the name of the directory on the server to which the system will transfer the file.
  - › To send the core dump file to NVS memory, use the **local** keyword.
- Example
 

```
host1(config)#exception dump 192.168.56.7 CORE_DUMPS
```
- Use the **no** version to disable the core dump.

**exception gateway**

- Use to specify the gateway through which the system sends the core dump file to the FTP server.
- Example
 

```
host1(config)#exception gateway 10.10.1.15
```
- Use the **no** version to restore the IP address to the null default value.

**exception protocol ftp**

- Use to set a user name and password for FTP access to the server where you transferred a core dump file. The default settings are the username anonymous and no password.
- Specify the number 8 before the user name and before the password to encrypt these values. By default, the user name and password are not encrypted.
- Example
 

```
host1(config)#exception protocol ftp 8 user_core 8
user_password
```
- Use the **no** version to restore the default settings.

### ***exception source***

- Use to set the IP address and mask of the system interface over which you want to send the core dump file to the FTP server.
- You can optionally include an IP address mask.
- Example

```
host1(config)#exception source 192.168.1.33 255.255.255.0
```

- Use the **no** version to restore the IP address and mask to the default null values.

### ***reload***

- Use to reload the software on the router immediately.
- Reloads the system software (.rel) file and the configuration (.cnf) file on the router.
- Example

```
host1#reload
```

- There is no **no** version.

### ***show exception dump***

- Use to display the parameters associated with the core dump operation.
- Field descriptions
  - › Dump host IP address – address of the host where the system is configured to transfer the dump file
  - › Dump directory – name of directory on the host where the system is configured to transfer the dump file
  - › Dump protocol – protocol used to send the core dump file; currently only FTP is supported
  - › User name – name configured for access to the core dump file on the FTP server
  - › Password – password configured for access to the core dump file on the FTP server
  - › Interface IP address – address of the system interface configured to send the core dump file
  - › Interface netmask – mask of the system interface configured to send the core dump file
  - › Gateway IP address – address of gateway configured between the system and the FTP server
- Example

```
host1#show exception dump  
Dump host IP address: 192.168.56.7  
Dump directory: CORE_DUMPS/  
Dump protocol: FTP  
User name: user_name
```

```

Password: user_password
Interface IP address:
Interface netmask:
Gateway IP address:
  
```

### Accessing the Core Dump File

If a module fails and saves a core dump file to NVS memory (which can take several minutes), you must transfer the file to a network host to examine it. You can transfer the core dump file when the module is back online or has assumed a redundant status. For information about the status of modules, see *ERX Installation and User Guide, Chapter 8, Troubleshooting*. To transfer the core dump file to a network host, use the **copy** command.

In a system with two SRP modules, the following behavior applies if you have configured the SRP modules to save core dump files to an FTP server:

- If the primary SRP module fails, it saves the core dump file to the FTP server before the standby SRP module assumes control.
- If the standby SRP module fails, it must save the core dump file to NVS because it has no access to any configured network host.

The **show version** command output indicates the failed SRP module state as “not responding” during the save process. Consequently, when the failed SRP module recovers and assumes the role of redundant module, the **show version** command output indicates the SRP module state as “standby.” You can now transfer the core dump file to a network host for examination. For example, to transfer the file *SRP\_1\_SC\_05\_24\_2000\_02\_20.dmp* from NVS of the failed SRP module to the host server1, enter the following command:

```

host1#copy SRP_1_SC_05_24_2000_02_20.dmp
host:/public/server1/SRP-5G_1_SC_05_24_2000_02_20.dmp
  
```

### **copy**

- Use to copy a core dump file.
- You cannot use wildcards.
- The file can be either a local or network file.
- You cannot create or copy over files generated by the system; however, you can copy such files to an unreserved filename.

#### • Example

```

host1#copy fault.dmp host:/public/server1/fault.dmp
  
```

- There is no **no** version.

### Understanding the Core Dump File

The dump file indicates which module has failed by referencing that module's hardware slot number. The hardware slot number is the slot number designation on the systems's backplane. This slot number is different from the chassis slot number that appears on the front of the chassis and in screen displays (for example, in the display resulting if you issue the **show version** command). Table 4-6 shows how the chassis slot numbers relate to the hardware slot numbers.

**Table 4-6** Chassis slot numbers vs. hardware slot numbers

ERX-700 series Hardware Slot Number	Slot Number on Chassis	ERX-1400 series Hardware Slot Number
1	0	0
3	1	1
4	2	2
5	3	3
6	4	4
7	5	5
8	6	7
	7	9
	8	10
	9	11
	10	12
	11	13
	12	14
	13	15

### Monitoring the System

---

This section provides basic system commands that allow you to display information about the system's state. The **show configuration** command, for example, allows you to display the system's entire configuration.

### **baseline show-delta-counts**

- Use to configure the system to always display statistics relative to the most recent appropriate baseline.
- The system collects many statistics during its operation. Various **show** commands are available to display these statistics. Baselining allows the user to identify a point in time relative to which such statistics can be reported.
- Typically, the optional **delta** keyword is used with **show** commands to specify that baselined statistics are to be shown. This command applies the “delta” function implicitly.
- Example

```
host1#baseline show-delta-counts
```
- Use the **no** version to have access to the total statistics.

### **show configuration**

- Use to display the system’s current configuration.
- You can create a configuration script from the output by saving it as a file with the .scr extension.
- You can exclude information about a particular type of interface.
- This command was formerly documented as **show config**; that abbreviation is still supported.
- You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. See *Chapter 2, Command Line Interface*, for details.
- This command is available only if the system is in Automatic Commit mode.
- Example - see the description on page 4-12.

### **show environment**

- Use to display information on the system’s physical environment, such as voltage or temperature.
- Optionally, specify the **all** keyword to view both the system environment information and the detailed temperature status table, or specify the **table** keyword to view only the temperature status table.
- The system displays a message if the voltage or temperature exceeds normal operating limits.
- The system enters thermal protection mode if the temperature exceeds maximum operating limits. For information about thermal protection mode, see *ERX Installation and User Guide, Chapter 8, Troubleshooting*.
- Field descriptions
  - › chassis – number of slots, midplane identifier, and hardware revision number
    - midplaneId14Slot – 5 Gbps, 14 slot midplane
    - midplaneId7Slot – 5 Gbps, 7 slot midplane

- midplaneIdRx1400 – 10 Gbps ASIC compatible, 12 line card slots, 2 SRP slots for ERX-1400 series
- midplaneIdRx700 – 10 Gbps ASIC compatible, 5 line card slots, 2 SRP slots for ERX-700 series
- › fabric – capacity and hardware revision of fabric
- › fans – status of fans
- › nvs – capacity of NVS and amount of space used
- › power – states of power feeds
- › srp redundancy – availability of a redundant SRP card
- › slots: cards missing or offline – status of each slot
  - online
  - standby
  - offline
  - empty
- › line redundancy – number of redundancy groups installed
  - width – number of slots the redundant midplane covers
  - spare – slot that contains spare line module
  - primary – slot that contains the primary line module
- › temperature – status of system temperature
- › timing – source of the timing signal
  - primary – type and status of the primary timing signal
  - secondary – type and status of the secondary timing signal
  - tertiary – type and status of the tertiary timing signal
  - auto-upgrade – status of the **auto-upgrade** parameter, which enables the system to revert to a higher-priority timing source after switching to a lower-priority timing source.
- › system operational – status of system
- › slot – number of the slot in which the module resides
- › processor temperature – temperature of line or SRP module
- › processor temperature status – temperature condition of the line module
  - normal – temperature is in normal range
  - too hot – module is too hot; system will go into thermal protection mode if temperature of any module exceeds 80° C
  - too cold – module is too cold; system will go into thermal protection mode if temperature of any module drops below –5° C
- › IOA temperature – temperature of corresponding I/O module
- › IOA temperature status – temperature condition of the corresponding module
  - normal – temperature is in normal range
  - too hot – module is too hot; system will go into thermal protection mode if temperature of any module exceeds 80° C
  - too cold – module is too cold; system will go into thermal protection mode if temperature of any module drops below –5° C

- Example

```

host1#show environment all
  chassis: 14 slot (id 0x3, rev. 0x0)
  fabric: 5 Gbps (rev. 1)
  fans: ok
  nvs: ok (81MB flash disk, 54% full)
  power: A ok, B not present
  srp redundancy: none
*** slots: cards missing or offline
    online: 6 9
    standby: 8
    offline: 2
    empty: 0 1 3 4 5 7 10 11 12 13
  line redundancy: 1 redundancy group(s)
    width 6, spare 8, primary 9
  temperature: ok
  timing: primary
    primary: internal SC oscillator (ok)
    secondary: internal SC oscillator (ok)
    tertiary: internal SC oscillator (ok)
    auto-upgrade enabled
*** system operational: no

          processor      processor      IOA          IOA
          temperature    temperature    temperature    temperature
slot (10C - 70C)      status      (10C - 70C)      status
----  -
0          31          normal          30          normal
3          31          normal          30          normal
5          31          normal          30          normal
7          31          normal          30          normal
processor temperature ranges
  below -5C is too cold
  above 80C is too hot
  low temperature warning below 10C
  high temperature warning above 70C
IOA temperature ranges
  below -5C is too cold
  above 80C is too hot
  low temperature warning below 10C
  high temperature warning above 70C

```

### ***show hosts***

- Use to display a list of configured network servers.
- Field Descriptions
  - › Static Host Table – information about the connected static hosts
    - name – name of the host
    - ip address – IP address of the host
    - type of host – type of host, for example ftp means an FTP server
- Example

```
host1#show hosts
      Static Host Table
      -----
name          ip address      type
----          -
host1         10.2.0.124      ftp
```

### ***show processes***

- Use to show amount of resources used by the system processes.
- Use no keywords or use the **cpu** keyword to display the CPU utilization.
- Use the **memory** keyword to display amount of memory used. Field descriptions
  - › name – name of process
  - › bytes allocated – bytes of memory allocated to the process
  - › bytes free – bytes of memory freed by the process, regardless of who originally allocated it
  - › blocks allocated – amount of memory currently allocated to the process
  - › blocks free – amount of memory freed by the process
  - › max free block – number of bytes in the one largest free block
  - › task name – name of process
  - › times invoked – number of times process has been invoked
  - › invocations per second – frequency of process invocation
  - › total running time (msec) – time the process has been running
  - › percent running time – percentage of total running time attributable to this process
  - › average time per invocation (usec) – average number of microseconds per invocation of this process
  - › 5 second utilization (%) – CPU utilization by process for the last 5 seconds
  - › 1 minute utilization (%) – CPU utilization by process for the last minute
  - › 5 minute utilization – CPU utilization by process for the last 5 minutes

- Examples

host1#show processes memory

Heap Statistics  
 -----

name	bytes		blocks		max
	allocated	free	allocated	free	free block
system	110430808	84680416	5284	256	84530744
file system	207600	303776	621	145	7216
Crldp.osHeap 1	10120	252008	9	1	252008
IpTemplateMgr General	104	2097032	2	2	2096992
Lsm.osHeap	134872	127256	21	2	127216
Rsvp.networkBuffers 1	0	32752	0	1	32752
Rsvp.osHeap 1	637536	411000	14	2	364832
.					
.					
.					
radius-rx*	0	39984	0	1	39984
radius-tx*	0	39984	0	1	39984
rip *	0	32752	0	1	32752
router buffer	0	524272	0	1	524272
ssscHeap	0	2097136	0	1	2097136

host1#show processes cpu

Process Statistics  
 -----

task name	times invoked	invocations per second	total	percent
			running time (msec)	running time
aaaAtm1483Config	1	0	0	0%
aaaServer	52	0	260	0%
agent1	399	0	3600	0%
arlEthHelp	362856	4	590	0%
.				
.				
.				
templateMgr	48	0	540	0%
timerd	2346566	32	0	0%
~GONE~	405202	5	184700	0%
~IDLE~	0	0	360	0%
~INTERRUPT~	8840490	121	51050	0%

task name	average			
	time per invocation (usec)	5 second utilization (%)	1 minute utilization (%)	5 minute utilization (%)
aaaAtml483Config	0	0	0	0
aaaServer	5000	0	0	0
agent1	9022	0	0	0
arlEthHelp	1	0	0	0
.				
.				
.				
templateMgr	11250	0	0	0
timerd	0	0	0	0
~GONE~	455	0	1	0
~IDLE~	---	0	0	0
~INTERRUPT~	5	0	0	0

**show reboot-history**

- Use to display the history of system and module resets.
- You can display the current reboot.hty file or a saved reboot history file.
- If you have a redundant system, it can be convenient to copy the redundant module's reboot.hty file to another filename for viewing with this command.
- Field descriptions
  - › Entry – number of entry in reboot history; numbers range from lowest (most recent reset) to highest (oldest reset)
  - › time of reset – timestamp for reset
  - › run state – state of system at reset
  - › image type – type of image running when the record is written
    - boot – module is running the boot file
    - diagnostics – module is running the diagnostics file
    - application – module is running the software file
  - › location – slot that reset
  - › build date – build date of software version
  - › reset type – cause of reset
- Example

```

host1#show reboot-history
*** Entry 1 ***
time of reset: TUE APR 10 2001 20:25:59 UTC
run state: unknown
image type: diagnostics
location: slot (7)
build date: 0x3abf4337 MON MAR 26 2001 13:25:11 UTC

```

```
reset type: user reboot, task "scheduler", reason "not specified"
*** Entry 2 ***
time of reset: TUE APR 10 2001 20:25:44 UTC
run state: unknown
image type: diagnostics
location: slot (8)
build date: 0x3abf5d5f MON MAR 26 2001 15:16:47 UTC
reset type: user reboot, task "scheduler", reason "not specified"
*** Entry 3 ***
time of reset: TUE APR 10 2001 20:25:03 UTC
run state: unknown
image type: diagnostics
location: slot (4)
build date: 0x3abf3ee0 MON MAR 26 2001 13:06:40 UTC
reset type: user reboot, task "scheduler", reason "not specified"
```

### ***show version***

- Use to display the configuration of the system hardware and the software version.
- Field descriptions
  - › Model identification
  - › Copyright – copyright details for the system software
  - › System Release – filename, version, and date of the system software currently running on the system
  - › System running for – time elapsed since the last boot of the system, date and time of last boot
  - › slot – physical slot that contains the line module
  - › state – status of the line module
    - booting – line module is booting
    - disabled (assessing) – system is evaluating the status of this line module
    - disabled (admin) – line module disabled via **slot disable** command
    - disabled (cfg error) – use of the line module in this slot violates the permitted configuration for the system. For example, the fabric cannot supply sufficient bandwidth to the line module in this position.
    - disabled (image error) – software for this line module is missing or corrupted
    - disabled (mismatch) – line module in this slot is a different type from that specified in the software. Correct the condition by inserting the original module, or use the **slot accept** command to find information about the new module.
    - hardware error – line module has a hardware fault
    - inactive – either the I/O module is not present, or this primary line module is fully booted and ready to resume operation. In the latter case, the spare is currently providing services.

- initializing – transitional state before the line module proceeds to the online, standby, or inactive state; diagnostics are complete, module is initializing software
- online – line module is operating
- not present – line module configured for this slot is missing
- not responding – line module has a hardware or ROM problem
- standby – spare line module or SRP module is fully booted and ready to operate if the primary line module or active SRP module fails
- unknown – transitional state while the SRP is initializing
- › type – kind of module; an “e” at the end of an SRP module type (for example, SRP-5Ge) indicates that the module includes error checking code (ECC)
- › admin – status of the slot in the software
  - enabled – slot is enabled
  - disabled – slot is disabled
- › spare – line module is a spare for line module redundancy
- › running release – software that is running on the line module
- › The following symbols and notices may be displayed at the end of the report:
  - # This release is a result of a subsystem override
  - \* This release is a result of a “boot slot” override
  - # The running or armed release on the slot is the same as the armed release for a subsystem. A subsystem is all the line modules of one type, such as OC3.
  - \* This release reflects whichever release the system is armed with at startup.
- › slot uptime – length of time for which the module has been operational; a value of --- indicates that the module is not available.
- Example

```

host1#show version
Juniper Networks Edge Routing Switch ERX-700
Copyright (c) 1999-200X Juniper Networks, Inc. All rights reserved.
System Release: rx x-y-z.rel
      Version: x.y.z (November 28, 200X 10:22)
System running for: 17 days, 17 hours, 15 minutes, 25 seconds
      (since THU NOV 30 200X 03:44:36 UTC)

              running
slot state  type   admin  spare  release  slot uptime
-----
0  online SRP-10G enabled  ---  mc_341.rel 0d00h:12m:52s
1  ---    ---    ---    ---    ---    ---
2  ---    ---    ---    ---    ---    ---
3  online CT3    enabled  ---  mc_341.rel 0d00h:12m:33s
4  ---    ---    ---    ---    ---    ---
5  online GE    enabled  ---  mc_341.rel  ---
6  ---    ---    ---    ---    ---    ---

```