Cisco ASR 5000 Series System Administration Guide

Version 9.0

Last Updated September 30, 2010

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-22969-02

Cisco ASR 5000 Series System Administration Guide

CONTENTS

About this Guide

This document pertains to features and functionality that run on and/or that are related to the Cisco® ASR 5000 Chassis, formerly the Starent Networks ST40.

Conventions Used

The following tables describe the conventions used throughout this documentation.

Icon	Notice Type	Description
	Information Note	Provides information about important features or instructions.
	Caution	Alerts you of potential damage to a program, device, or system.
	Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.
	Electro-Static Discharge (ESD)	Alerts you to take proper grounding precautions before handling a product.

Typeface Conventions	Description
Text represented as a `screen display`	This typeface represents displays that appear on your terminal screen, for example: `Login:`
Text represented as **`commands`**	This typeface represents commands that you enter, for example: **`show ip access-list`** This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.
Text represented as a **`command`** *`variable`*	This typeface represents a variable that is part of a command, for example: **`show card`** *`slot_number`* slot_number is a variable representing the desired chassis slot number.
Text represented as menu or sub-menu names	This typeface represents menus and sub-menus that you access within a software application, for example: Click the **File** menu, then click **New**

Command Syntax Conventions	Description
{ **`keyword`** or *`variable`* }	Required keywords and variables are surrounded by grouped brackets. Required keywords and variables are those components that are required to be entered as part of the command syntax.

Command Syntax Conventions	Description
[**keyword** or *variable*]	Optional keywords or variables, or those that a user may or may not choose to use, are surrounded by square brackets.
|	With some commands there may be a group of variables from which the user chooses one. These are called alternative variables and are documented by separating each variable with a vertical bar (also known as a pipe filter). Pipe filters can be used in conjunction with required or optional keywords or variables. For example: **{ nonce | timestamp }** OR [**count** *number_of_packets* | **size** *number_of_bytes*]

Contacting Customer Support

Use the information in this section to contact customer support.

For New Customers: Refer to the support area of http://www.cisco.com for up-to-date product documentation or to submit a service request. A valid username and password is required to this site. Please contact your local sales or service representative for additional information.

For Existing Customers with support contracts through Starent Networks: Refer to the support area of https://support.starentnetworks.com/ for up-to-date product documentation or to submit a service request. A valid username and password is required to this site. Please contact your local sales or service representative for additional information.

Important: For warranty and repair information, please be sure to include the Return Material Authorization (RMA) tracking number on the outside of the package.

Chapter 1
Command Line Interface Overview

This chapter describes the numerous features in the command line interface (CLI). Included is information about the architecture of the CLI, its command modes and user privileges, how to obtain help within the CLI, and other key items.

The operating system provides the software that controls the overall system logic, control processes, and the CLI. The CLI is a multi-threaded user interface that allows you to manipulate, configure, control, and query the hardware and software components that make up the system and its hosted services. In addition, the CLI can host multiple instances of management and service configuration sessions. This allows multiple users to simultaneously access and manage multiple hosted services.

This section provides the following information about the CLI:

CLI Structure

CLI commands are strings of commands or keywords and user-specified arguments that set or modify specific parameters of the system. Commands are grouped by function and the various command modes with which they are associated.

The structure of the CLI is hierarchical. All users begin at a specific entry point into the system, called the Exec (Execute) Mode, and then navigate through the CLI according to their defined user privileges (access level) by using other command modes.

CLI Command Modes

There are two primary CLI command modes:

- **Exec (Execute) Mode:** The Exec mode is the lowest level in the CLI. The Exec mode is where you execute basic commands such as show, and ping. When you log into the CLI, you are placed in this mode by default.
- **Config (Configuration) Mode:** The Config mode is accessible only by users with administrator and security administrator privileges. If you are an administrative user, in this mode you can add and configure contexts and access the configuration sub-modes to configure protocols, interfaces, ports, services, subscribers, and other service-related items.

As explained above, the entry point into the CLI is called Exec Mode. In the initial CLI login, all users are placed into the default local context, which is the CLI's default management context. From this context, administrative users can access the Config Mode and define multiple service contexts.

Refer to the mode entry-path diagrams at the beginning of each mode chapter in the *Command Line Interface Reference*.

Important: The commands or keywords/variables that are available are dependent on platform type, product version, and installed license(s).

CLI Administrative Users

This section contains information on the administrative user types and privileges supported by the system.

Administrative User Types

There are two types of administrative users supported by the system:

- **Context-level administrative users:** This user type is configured at the context-level and relies on the AAA subsystems for validating usernames and passwords during login. This is true for both administrative user accounts configured locally through a configuration file or on an external RADIUS server. Passwords for these user types are assigned once and are accessible in the configuration file.
- **Local-users:** This user type provides support for ANSI T1.276-2003 password security protection. Local-user account information, such as passwords, password history, and lockout states, is maintained in non-volatile memory on the CompactFlash module and in the Shared Configuration Task (SCT). This information is maintained in a separate file, not in configuration files used by the system. As such, the configured local-user accounts are not visible with the rest of the system configuration.

Local-user and context-level administrative accounts can be used in parallel. However, a mechanism is provided to de-activate context-level administrative user accounts thereby providing access only to local-user accounts.

Authenticating Administrative Users with RADIUS

To authorize users via RADIUS, you must include two RADIUS attributes in the RADIUS Access-Accept message:

- RFC 2865 standard Service-Type
- Starent Vendor-Specific Attribute (VSA) SN1-Admin-Permission.

The default permission is none (0), meaning that service is refused even if properly authenticated via RADIUS.

RADIUS Mapping System

RADIUS server configuration depends on the type of server used and the instructions distributed by the server manufacturer. The following table shows the supported attribute/value mapping system that is constant, regardless of server manufacturer or model:

Table 1. RADIUS Attribute/Value Mapping System

Attribute	Value

Attribute	Value
Framed	2
Administrative (Administrator)	6
NAS_Prompt	7
Authenticate_Only	8
Authorize_Only	17
Inspector	19650516
Security_Admin	19660618

RADIUS Privileges

There are four RADIUS privilege roles. The following table shows the relationship between the privilege roles in the CLI configuration and RADIUS Service-Type.

Table 2. CLI Privilege Roles and RADIUS Service Types

CLI Configuration Parameter	RADIUS Service Type	Show Admin Type
administrator	Security_Admin (19660618)	admin
config_administrator	Administrative (6)	cfgadm
operator	NAS_Prompt (7)	oper
inspector	Inspector (19650516)	inspect

Administrative User Privileges

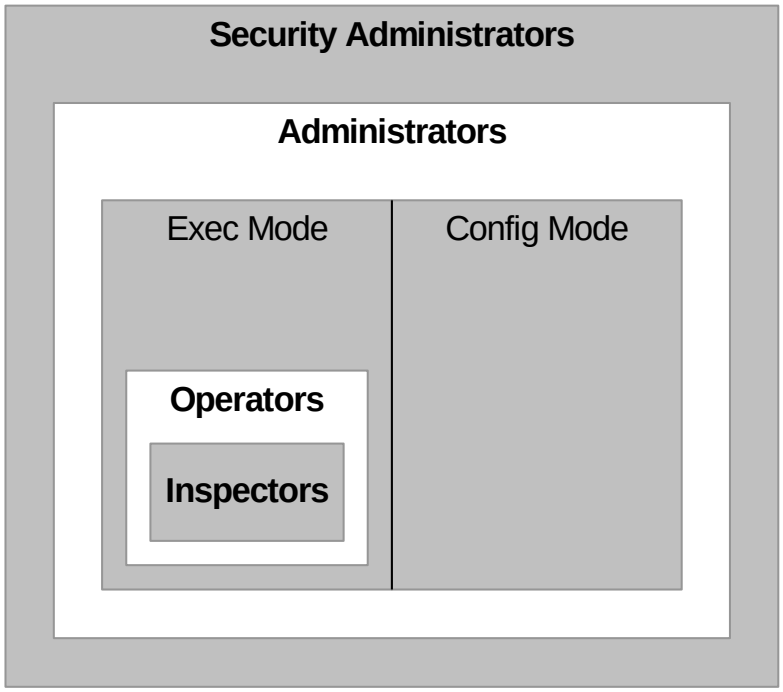
Regardless of the administrative user type, the system supports four user privilege levels:

- **Inspector**: Inspectors are limited to a small number of read-only Exec Mode commands. The bulk of these are show commands for viewing a variety of statistics and conditions. The Inspector cannot execute show configuration commands and does not have the privilege to enter the Config Mode.
- **Operator**: Operators have read-only privileges to a larger subset of the Exec Mode commands. They can execute all commands that are part of the inspector mode, plus some system monitoring, statistic, and fault management functions. Operators do not have the ability to enter the Config Mode.
- **Administrator**: Administrators have read-write privileges and can execute any command in the CLI except for a few security-related commands that can only be configured by Security Administrators. Administrators can configure or modify system settings and can execute all system commands, including those available to the Operators and Inspectors.

- **Security Administrator**: Security Administrators have read-write privileges and can execute all CLI commands, including those available to Administrators, Operators, and Inspectors.

The following figure represents how user privileges are defined in the CLI configuration modes.

Figure 1. User Privileges



Though the privilege levels are the same regardless of user type, the corresponding user type names differ slightly. The following table displays the privilege level to administrative user type mappings:

Table 3. User Privilege to User Type Mapping

User Type as Defined by T1.276-2003	Local-User Level User	Context-Level User
System Security Administrator	Security Administrator	Administrator
Application Security Administrator	Security Administrator	Administrator
System Administrator	Administrator	Config-Administrator
Application Administrator	Administrator	Config-Administrator
Application User/Operator	Operator	Operator
not applicable	Inspector	Inspector

Configure context-level administrative users in the Context Configuration Mode with the **`administrator`**, **`config-administrator`**, **`operator`**, and **`inspector`** commands.

Configure local-user administrative users at the Global Configuration Mode with the **`local-user username`** command.

You can further refine administrative levels to include access to certain features with the following feature-use administrative user options:

- **Lawful Intercept (LI) Administrative User:** To configure and manage LI-related issues, configure at least one administrative user account with LI functionality privileges.

Important: This privilege is available only for context-level administrative users. In addition, to ensure security in accordance with the standards, LI administrative users must access the system through the Secure Shell Protocol (SSH).

- **Enhanced Charging Service (ECS) Administrative User:** To log in and execute ECS-related commands, configure at least one administrative user account with ECS functionality privileges.

All system users can be configured within any context. However, it is recommended that you configure users in the system's management context called local. Refer to sections later in this chapter for additional information about contexts.

Allowed Commands per User Type

With the exception of security administrators, all other management users are limited to a subset of the entire command list as described in the *Command Line Interface Reference*. This section defines the commands allowed for each management user type. As stated previously, inspectors and operators are limited to only a subset of the Exec Mode commands.

Inspector Mode Commands

In the Exec mode, system inspectors can access the following commands:

- abort
- autoconfirm
- context
- crypto-group
- default terminal
- exit
- help
- logs checkpoint
- monitor subscriber
- no logging active

- no logging trace
- no reveal disabled commands
- no timestamps
- no autoconfirm
- ping
- reveal disabled commands
- show (except show snmp communities and show snmp transports)
- sleep
- start crypto security-association
- terminal length
- terminal width
- timestamps
- traceroute

Operator Mode Commands

In the Exec mode, system operators can access all inspector mode commands plus the following commands:

- aaa test
- alarm cutoff
- bulkstats force
- card
- clear (a subset of all clear command variations)
- debug
- dhcp test
- gtpc test
- gtpp interim
- gtpp test
- gtpu test
- gtpv0 test
- host
- logging active
- logging filter
- logging trace
- newcall
- no card
- no debug

- no newcall policy
- port
- ppp echo-test
- radius interim accounting
- radius test
- rlogin
- show access-group
- show access-list
- show access-flow
- show access statistics
- show configuration
- show snmp transports
- ssh
- telnet
- test alarm

Administrator Mode Commands

Administrators can access all system commands except:

Context Config Mode

- config-administrator
- operator
- inspector
- administrator

Global Config Mode

- snmp community
- snmp user
- local-user
- suspend local-user

Exec Mode

- show snmp communities
- clear (all clear command variations)
- show local-user
- password change local-user

Security Administrator Mode Commands

Security administrators can access all system commands.

CLI Contexts

A context is a group of configuration parameters that apply to the ports, interfaces, and protocols supported by the system. You can configure multiple contexts on the system, each of which resides as a separate, logically independent instance on the same physical device. The CLI can host multiple contexts within a single physical device. This allows wireless service providers to use the same system to support:

- Different levels of service
- Multiple wholesale or enterprise customers or customer groups
- Different classes of customers based on defined Class of Service (CoS) parameters
- IP address pools across multiple contexts, thus saving IP address allocation
- Enhanced security

Each defined context operates independently from any other context(s) in the system. Each context contains its own CLI instance, IP routing tables, access filters, compression methods, and other configured data.

By default, a single system-wide context called local, is used exclusively for the management of the system. Think of the local context as the root directory of the system, since you can define and access all other contexts from this point. You cannot delete the local context. From this location in the CLI, you can:

- Create and configure other service contexts that contain different service configurations
- Configure system-wide services such as CORBA and SNMP management interfaces, physical management ports, system messages, and others

Important: The system requires that you define at least one context in addition to the Local context. This isolates system management functions from application or service functions.

Administrative users add contexts through the Global Configuration Mode. A substantial advantage of configuring numerous service contexts is that it allows operators to broadly distribute different subscribers across the system. This greatly enhances the performance of the system and minimizes the loss of sessions should a failure occur.

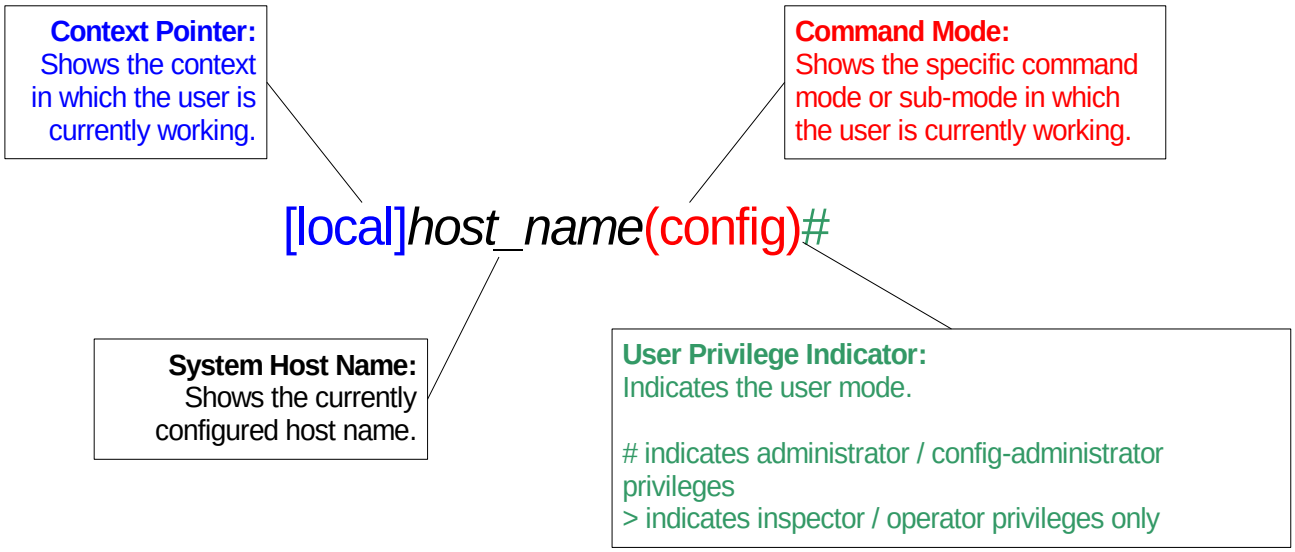
Understanding the CLI Command Prompt

The CLI provides an intuitive command prompt that informs you of:

- Exactly where you are located within the CLI
- The command mode you are using
- Your user privilege level.

The following figure shows the various components of the command prompt.

Figure 2. CLI Command Prompt



CLI Command Syntax

This section describes the components of the CLI command syntax that you should be familiar with prior to using the CLI. These include:

- **Commands**: Specific words that precede, or initiate, a specific function.
- **Keywords**: Specific words that follow a command to more clearly dictate the command's function.
- **Variables**: Alpha, numeric, or alphanumeric values that are user-supplied as part of the command syntax. Sometimes referred to as arguments, these terms further specify the command function.
- **Repetitive keywords (+)**: Specific keyword, that when followed by a plus (+) sign, indicates that more than one of the keywords can be entered within a single command.

Example

In the following example, *slot_number* is the command variable for the **info** keyword:

```
show slot infoslot_number
```

slot_number is a variable representing a particular slot (1 through 48).

Entering and Viewing CLI Commands

This section describes various methods for entering commands into the CLI.

Typing each command keyword, argument, and variable can be time-consuming and increase your chance of making mistakes. The CLI therefore, supports the following features to assist you in entering commands quickly and more accurately. Other features allow you to view the display and review previously entered commands.

Entering Partial CLI Commands

In all of the modes, the CLI recognizes partially-typed commands and keywords, as long as you enter enough characters for the command to be unambiguously recognized by the system. If you do not enter enough characters for the system to recognize a unique command or keyword, it returns a message listing all possible matches for the partial entry.

Example

If you enter the partial command **conf** and press <**Enter**>, you enter the Global Configuration Mode. If you were to enter only **co**, the system would respond with the message:

```
Ambiguous Command
```

CLI Command Auto-completion

Use the command auto-completion feature to automatically complete unique CLI commands. Press the <**Tab**> key after entering enough characters to enable this feature.

Example

[local]*host_name*# sho<Tab>
[local]*host_name*# show

If you do not enter enough characters to allow the CLI to determine the appropriate command to use, the CLI displays all commands that match the characters you entered with auto-completion:

Example

```
[local]host_name# sh<Tab>

show     shutdown

[local]host_name#
```

Enter a question mark (?) after a partial command to display all of the possible matching commands, and their related help text.

Example

```
[local]host_name# sh?

show - Displays information based on a specified argument
```

```
shutdown - Terminates execution of all tasks within the entire chassis

[local]host_name#
```

Using CLI Auto-Pagination

When you enter commands whose expected results exceed the terminal window's vertical display, the auto-pagination function pauses the display each time the terminal window reaches its display limit. Press any key to display the next screen of results.

By default, auto-pagination functionality is disabled. To enable auto-pagination, type the pipe command: **| more**

```
[local]host_name# show configuration | more
```

Important: When auto-pagination is enabled, if a command's output exceeds the terminal window's vertical display parameters, you can exit by entering "**q**". This returns you to the CLI prompt.

Using CLI Autoconfirmation

By default, the system is configured to prompt all administrative users with a confirmation prior to executing certain commands. This functionality serves two purposes:

- Helps ensure that you do not execute an unwanted configuration change.

Example
Saving a configuration:

```
[local]host_name# save configuration

Are you sure ? [Yes | No]:
```

- Indicates potential misspellings of names during configuration. The first time you configure an element name (context, subscribers, services, etc.), the prompt is displayed. The prompt is not displayed for subsequent entries of the name. Therefore, if you see the confirmation prompt after entering the name of a previously configured element, it is likely that you misspelled the name.

Examples
You create context named "newcontext":

```
[local]host_name(config)# context newcontext

Are you sure ? [Yes | No]: yes

[newcontext]host_name(config-ctx)#
```

You revisit the context named “newcontext”:

```
[local]host_name(config)# context newcontext
[newcontext]host_name(config-ctx)#
```

On another occasion, you misspell the context named “newcontext”:

```
[local]host_name(config)# context mewcontext
Are you sure ? [Yes | No]:n
Action aborted
[local]host_name(config)#
```

After aborting the above action, you can again revisit “newcontext”:

```
[local]host_name(config)# context newcontext
[newcontext]host_name(config-ctx)#
```

You can control CLI autoconfirmation at the following levels:

- **Specific administrative user sessions:** To enable or disable autoconfirmation, use the [no] autoconfirm commands while in the Exec mode.
- **All Future Sessions:** To disable or re-enable autoconfirmation for all future sessions, use the [no]autoconfirm commands while in the Global Config mode.
- **For specific commands:** Disable autoconfirmation for various commands that support the -noconfirm keyword, such as the save configuration or card reboot commands.

Regulating the Command Output

For many CLI commands, you can use **| grep** and/or **| more** keywords to regulate or control the command’s output.

Use the | grep keyword to filter through a command’s output for certain expressions or patterns. Only those portions of the output that contain or exclude the pattern are displayed. The | grep has the following syntax:

| grep [-i | -v | --ignore-case | --invert-match] *expression*

Table 4. grep Keywords

Alternative Keyword	Description
-i	Specifies the filtering of the command’s output for a particular expression while ignoring case. Lower case matches the same as upper case.
-v	Specifies the filtering of the command’s output for everything excluding a particular expression.
--ignore-case	The long form of the **-i** option.

Alternative Keyword	Description
--invert-match	The long form of the **-v** option.
expression	Specifies the character pattern to find in the command's output.

Use the **| more** keyword to pause the terminal each time the terminal window reaches its display limit. Press any key to display the next screen. The function of this keyword is identical to the **autoless** command, except that you must manually enter it on a command-by-command basis.

Viewing Command History

To view a history of all commands line by line, simply scroll up or down with the <**up arrow**> and <**down arrow**> cursor keys on the keyboard.

The operating system supports EMACS-style text editing commands. This standard UNIX text editor format allows you to use keyboard-based shortcut keys for maneuvering around the CLI. The following table lists these available shortcut keys.

Table 5. EMACS Shortcut Keystrokes

Shortcut Keys	Description
<Ctrl + p> and <up arrow>	Recalls previous command in the command history
<Ctrl + n> and <down arrow>	Recalls next command in the command history
<Ctrl + f> and <right arrow>	Moves cursor forward by one character in command line
<Ctrl + b> and <left arrow>	Moves cursor backward by one character in command line
<Esc> + <f>	Moves cursor forward by one word in command line
<Esc> + <b>	Moves cursor backward by one word in command line
<Ctrl> + <a>	Moves cursor to the beginning of the command line
<Ctrl> + <e>	Moves cursor to the end of the command line
<Ctrl> + <k>	Deletes the current command line from the insertion point to the end of the line
<Ctrl> + <u>	Deletes the current command line from the insertion point to the beginning of the line
<Ctrl> + <d>	Deletes a single character in the current command line
<Esc> + <d>	Deletes a word in the current command line
<Ctrl> + <c>	Quits editing the current line
<Ctrl> + <l>	Refreshes the display
<Ctrl> + <t>	Transposes (or switches) the two characters surrounding the insertion point

Obtaining CLI Help

The CLI provides context-sensitive help for every command token and keyword available to you. To obtain, use one of these methods:

- **Command Help:** Command help provides assistance for a specific command. Type a question mark (?) at the end of the specific command to accesses help.

Example

```
[local]host_name# test?

test - Performs test on followed mechanism
```

- **Keyword Help:** Keyword help provides assistance in determining the next keyword, argument, or option to use in the command syntax. Enter the command keyword, enter a space, and then type a question mark (?).

Example

```
[local]host_name# test alarm ?

audible - Tests internal audible alarm buzzer on SPC

central-office - Tests specified central office alarm relays on SPIO card

<cr> - newline
```

- **Variable Help:** Variable help provides the correct format, value, or information type for each variable that is part of the command syntax. For commands with variables, enter the command keyword, enter a space, and then type a question mark (?).

Example

```
[local]host_name# show card info ?

<Enter card number as an integer ranging 1 to 48> | - Pipeline <cr> -
Carriage Return or <Enter> key
```

Exiting the CLI and CLI Command Modes

A CLI session is defined as the successful login into the CLI. When you establish a CLI session, you are placed into the system's Exec Mode. Depending upon your user privilege level, you can:

- Use the *local* context to perform system management functions
- Move to an assigned context and work in Exec Mode
- Move to an assigned context as an administrative user and work in Global Configuration Mode or other configuration sub-mode

This section addresses how to properly exit the various modes and the CLI.

Exiting Configuration Sub-modes

To exit a configuration sub-mode and return to the next highest configuration sub-mode or Global Configuration Mode, type the exit command at the system prompt.

Example

```
[context_name]host_name(config-ctx)# exit

[local]host_name(config)#
```

Important: The CLI supports implicit mode-exits when using configuration files. Therefore, configuration files do not have to contain all of the required exit commands for you to leave various sub-config modes.

To exit a sub-mode and return to the Exec Mode, enter the **end** command.

Example

```
[local]host_name(config-ctx)# end

[local]host_name#
```

Exiting Global Configuration Mode

To exit Global Configuration Mode, and return to the Exec Mode prompt, type the **exit** command at the prompt.

Ending a CLI Session

To end a CLI session and exit the CLI, type the **`exit`** command at the *local* Exec Mode prompt.

Accessing the CLI

Access the CLI through the following methods:

- Local login through a Console port using the RS-232 serial cable supplied with the card
- Remote login using Telnet and Secure Shell (SSH) access to the CLI through any IP interface on the system.

Important: Even though you can access the CLI remotely through any available IP interface, it is recommended that management traffic be isolated from network traffic by using one of the SPIO card management interfaces. You can use remote login methods only after the system has been configured to support the various access methods.

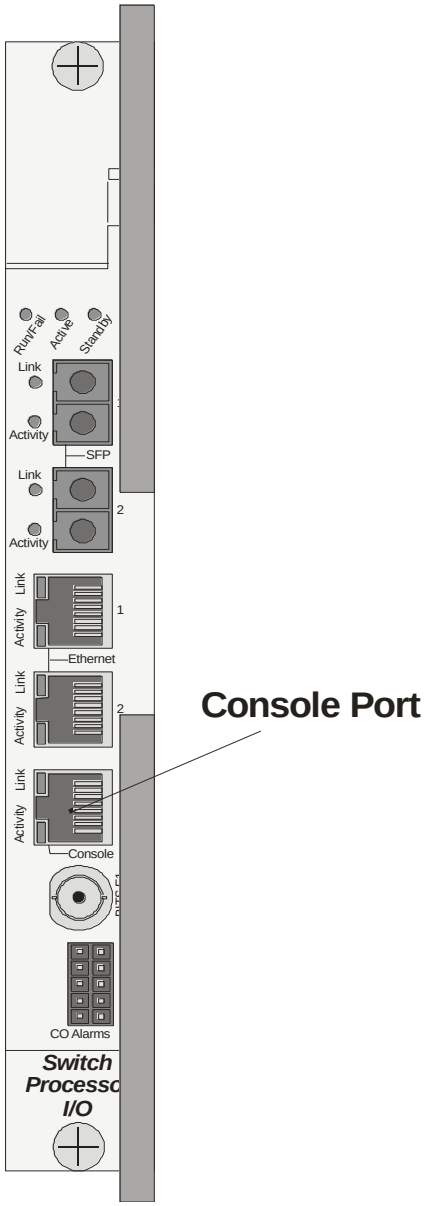
Important: Multiple CLI sessions are supported, but the number of sessions is dependent on the amount of available memory. The Resource Manager reserves enough resources so that as a minimum, seven CLI sessions are assured. One of the CLI sessions is always reserved for use exclusively by a CLI session on an SPIO console interface. Additional CLI sessions beyond the pre-reserved set are permitted if sufficient SPC resources are available. If the Resource Manager is unable to reserve additional resources, you are prompted whether to allow the system to create the new CLI session, even without the reserved resources.

Important: Multiple CLI sessions are supported, but the number of sessions is dependent on the amount of available memory. The Resource Manager reserves enough resources so that as a minimum, 15 CLI sessions are assured. One of the CLI sessions is always reserved for use exclusively by a CLI session on an SPIO console interface. Additional CLI sessions beyond the pre-reserved set are permitted if sufficient SMC resources are available. If the Resource Manager is unable to reserve additional resources, you are prompted whether to allow the system to create the new CLI session, even without the reserved resources.

Accessing the CLI Locally Using the Console Port

This section provides instructions for accessing the CLI locally through the console port.

Figure 3. Console Port



Access the console port with the RJ-45-to-DB-9 serial (EIA-232) cable that is shipped with the Switch Processor Input/Output (SPIO)Connect to a workstation that has a communications application that accesses the workstation's serial port, such as Minicom for Linux or HyperTerminal® for MicroSoft Windows®.

Each of the two SPIO Line Cards installed in the system provides a console port for accessing the CLI. The CLI is only accessible from the SPIO that is active—typically the SPIO installed in chassis slot 24.

For normal operation, the SPC or SMC in chassis slot 8 serves as the active processing card for the system. The SPIO that corresponds to this SPC or SMC is installed in slot 24. For the processing card in chassis slot 9, the corresponding SPIO is installed in slot 25.

Important: In the event of an SPC or SMC switchover, in which processes are switched from the processing card in slot 8 that was previously active to the redundant processing card in slot 9, the SPIO in slot 24 continues to serve as the active SPIO. Therefore, the console port is still accessible through that SPIO.

Follow the instructions below to connect to the console port.

1. Connect the RJ-45 end of the cable to the port labeled *Console*.
2. Connect the DB-9 end of the cable to the serial port on the workstation.
3. Configure the communications application to support the following:

Parameter	Setting
Baud Rate	115,200 bps
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

Important: To change the configuration defined in the table above, modify the `terminal` command located in the Global configuration mode.

4. At the terminal window, press **Enter**.
5. If no configuration file is present (that is, this is the first time the system is powered), the CLI prompts you as to whether or not you want to use the Quick Setup Wizard. If the system was configured previously, you are prompted to enter a username and password.

Remotely Accessing the CLI

To remotely access the CLI through a defined management interface, you must first configure the remote access method (such as Telnet or SSH).

You can find examples of how to configure this in the *Getting Started* chapter.

Chapter 2
Understanding System Operation and Configuration

The system provides wireless carriers with a flexible solution that can support a wide variety of services. These services are described in detail in the *System Overview Guide*.

Prior to connecting to the command line interface (CLI) and beginning the configuration, there are important things to understand about how the system supports these applications. This chapter provides terminology and background information that must be considered before attempting to configure the system. The following sections are included:

Terminology

This section defines some of the terms used in the following chapters.

Contexts

A context is a logical grouping or mapping of configuration parameters that pertain to various physical ports, logical IP interfaces, and services. A context can be thought of as a virtual private network (VPN).

The system supports the configuration of multiple contexts. Each is configured and operates independently from the others. Once a context has been created, administrative users can then configure services, logical IP interfaces, and subscribers for that context. Administrative users then bind the logical interfaces to physical ports.

Contexts can also be assigned domain aliases; if a subscriber's domain name matches one of the configured alias names for that context, then that context is used.

Ports

Ports are the physical Ethernet connectors (Ethernet 10/100, Ethernet 1000 Line Cards, four-port Quad Gigabit Line Cards (QGLC), and 10-Gigabit Ethernet Line Cards (XGLC) . Ethernet port configuration addresses traffic profiles, data encapsulation methods, media type, and other information for physical connectivity between the system and the rest of the network. Ports are identified by the chassis slot number for the line card, followed by the physical connector number.

For example, Port 24/1 identifies connector number 1 on the card in slot 24.

Associate ports with contexts through bindings. For additional information on bindings, refer to the *Bindings* section below. You can configure each physical port to support multiple logical IP interfaces, each with up to 17 IP addresses (one primary and up to 16 secondaries).

Logical Interfaces

You must associate a port with a virtual circuit or tunnel called a *logical interface* before the port can allow the flow of user data. A logical interface within the system is defined as the logical assignment of a virtual router instance that provides higher-layer protocol transport, such as Layer 3 IP addressing. Interfaces are configured as part of the VPN context and are independent from the physical port that will be used to bridge the virtual interfaces to the network.

There are several types of logical interfaces to configure to support Simple and Mobile IP data applications.

Management interface: This interface provides the point of attachment to the management network. The interface supports remote access to the CLI It also supports Common Object Request Broker Architecture (CORBA)-based management via the Web Element Manager application, and event notification via the Simple Network Management Protocol (SNMP).

Define management interfaces in the *local* context and bind them to the ports on the Switch Processor Input/Output (SPIO) cards.

Bindings

A binding is an association between elements within the system. There are two types of bindings: static and dynamic.

- Static binding is accomplished through system configuration. Static bindings associate:
  - A specific logical interface (configured within a particular context) to a physical port. Once the interface is bound, traffic can flow through the context as if it were any physically-defined circuit. Static bindings support any encapsulation method over any interface and port type.
  - A service to an IP address assigned to a logical interface within the same context. This allows the interface to take on the characteristics (that is, support the protocols) required by the service.
- Dynamic binding associates a subscriber to a specific egress context based on the configuration of their profile or system parameters. This provides a higher degree of deployment flexibility, as it allows a wireless carrier to support multiple services and facilitates seamless connections to multiple networks.

Services

Configure services within a context to enable certain functionality. The following are examples of services you can configure on the system, subject to licensing availability and platform type:

- GGSN services
- SGSN Services
- PDSN services
- FA services
- HA services
- LAC services
- DHCP services
- ASN GW services
- ASN Paging Controller and Location Registry services
- PDIF services
- SCM services (P-CSCF, S-CSCF, A-BG)
- Mobility Management Entity (MME) Services
- PDN Gateway (P-GW) Services
- Serving Gateway (S-GW) Services

- Home-NodeB Gateway (HNB-GW) Services

AAA Servers

For most configurations, AAA servers are used to store profiles, perform authentication, and maintain accounting records for each mobile data subscriber. The AAA servers communicate with the system over the AAA interface. The system supports the configuration of up to 128 AAA servers.

It is important to note that for Mobile IP, there can be foreign AAA (FAAA) and home AAA (HAAA) servers. The FAAA server(s) typically resides in the carrier's network. The HAAA server(s) could be owned and controlled by either the carrier or the home network. If the HAAA server is owned and controlled by the home network, accounting data is transferred to the carrier via a AAA proxy server.

Important: Mobile IP support depends on the availability and purchase of a standalone HA license or a license bundle that includes HA.

Subscribers

Subscribers are the end-users of the service; they gain access to the Internet, their home network, or a public network through the system. There are three primary types of subscribers:

- **RADIUS-based Subscribers:** The most common type of subscriber, these users are identified by their International Mobile Subscriber Identity (IMSI) number, an Electronic Serial Number (ESN), or by their domain name or user name. They are configured on and authenticated by a RADIUS AAA server.

  Upon successful authentication, various attributes that are contained in the subscriber profile are returned. The attributes dictate such things as session parameter settings (for example, protocol settings and IP address assignment method), and what privileges the subscriber has.

  Important: Attribute settings received by the system from a RADIUS AAA server take precedence over local-subscriber attributes and parameters configured on the system.

- **Local Subscribers**: These are subscribers, primarily used for testing purposes, that are configured and authenticated within a specific context. Unlike RADIUS-based subscribers, the local subscriber's user profile (containing attributes like those used by RADIUS-based subscribers) is configured within the context where they are created.

  When local subscriber profiles are first created, attributes for that subscriber are set to the system's default settings. The same default settings are applied to all subscriber profiles, including the subscriber named *default* which is created automatically by the system for each system context. When configuring local profile attributes, the changes are made on a subscriber-by-subscriber basis.

Important: Attributes configured for local subscribers take precedence over context-level parameters. However, they *could* be over-ridden by attributes returned from a RADIUS AAA server.

- **Management Subscribers**: A management user is an authorized user who can monitor, control, and configure the system through the command-line interface (CLI) or Web Element Manager application. Management is performed either locally, through the system console port, or remotely through the use of the Telnet or secure shell (SSH) protocols. Management users are typically configured as a local subscriber within the Local context, which is used exclusively for system management and administration. As with a local subscriber, a management subscriber's user profile is configured within the context where the subscriber was created (in this case, the Local context). However, management subscribers may also be authenticated remotely via RADIUS, if an AAA configuration exists within the local context.

How the System Selects Contexts

The previous section of this chapter defined what a context is and how it is used within the system. This section provides details about the process that determines which context to use for context-level administrative user or subscriber sessions. Understanding this process allows you to better plan your configuration in terms of how many contexts and interfaces you need configure.

Context Selection for Context-level Administrative User Sessions

The system comes configured with a context called local that you use specifically for management purposes. The context selection process for context-level administrative users (those configured within a context) is simplified because the management port(s) on the SPIO are associated only with the Local context. Therefore, the source and destination contexts for a context-level administrative user responsible for managing the entire system should always be the local context.

Although this is not commonly done, a context-level administrative user can also connect through other interfaces on the system and still have full system management privileges.

A context-level administrative user can be created in a non-local context. These management accounts have privileges only in the context in which they are created. This type of management account can connect directly to a port in the context in which they belong, if local connectivity is enabled (SSHD, for example) in that context.

For all FTP or SFTP connections, you must connect through a SPIO interface. If you SFTP or FTP as a non-local context account, you must use the username syntax of *username@contextname*.

The context selection process becomes more involved if you are configuring the system to provide local authentication or work with a AAA server to authenticate the context-level administrative user.

The system gives you the flexibility to configure context-level administrative users locally (meaning that their profile will be configured and stored in its own memory), or remotely on an AAA server. If a locally-configured user attempts to log onto the system, the system performs the authentication. If you have configured the user profile on an AAA server, the system must determine how to contact the AAA server to perform authentication. It does this by determining the AAA context for the session.

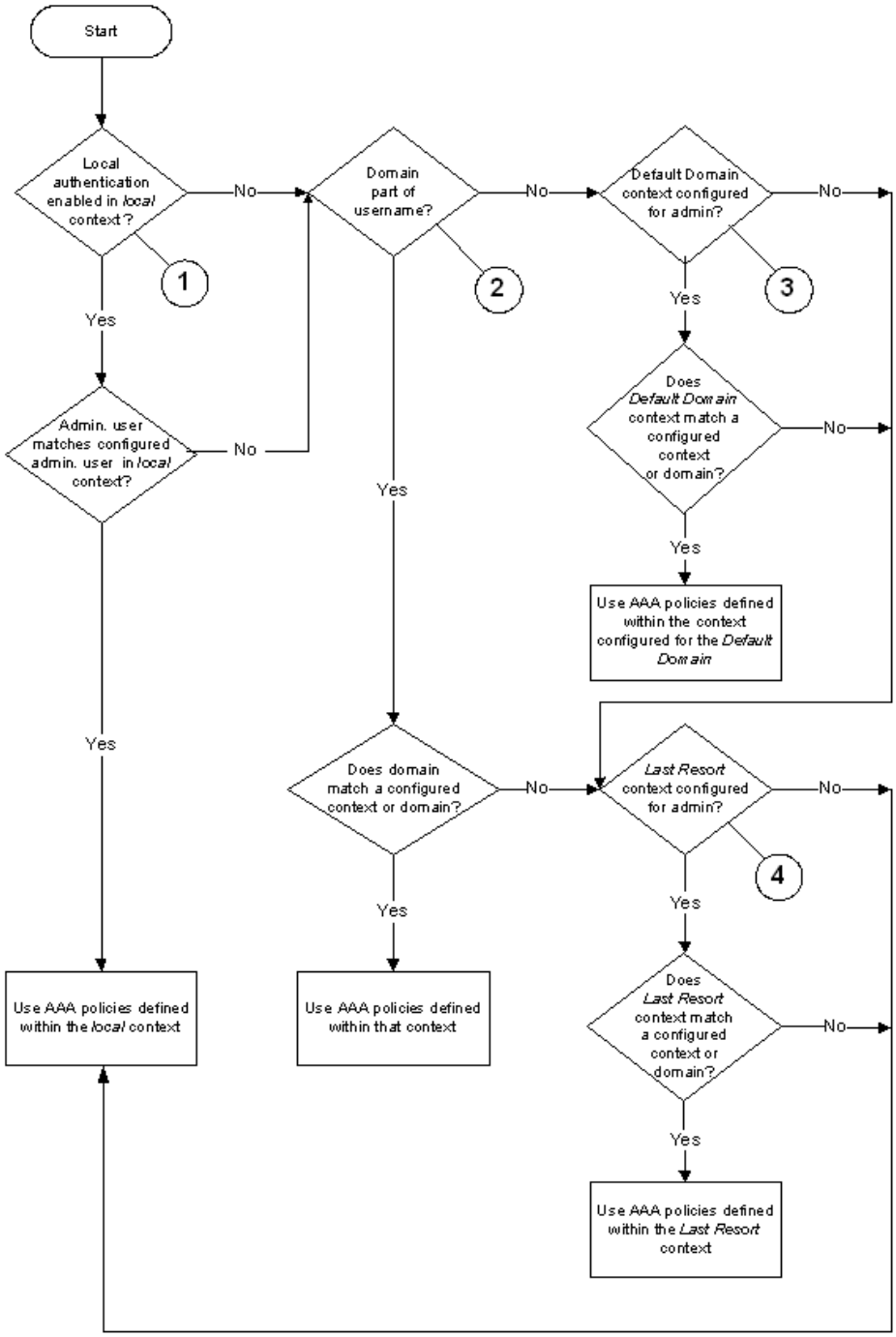
The following table and figure describe the process that the system uses to select an AAA context for a context-level administrative user.

Table 6. Context-level Administrative User AAA Context Selection

Item	Description
1	During authentication, the system determines whether local authentication is enabled in the *local* context. If it is, the system attempts to authenticate the administrative user in the *local* context. If it is not, proceed to item 2 in this table. If the administrative user's username is configured, authentication is performed by using the AAA configuration within the *local* context. If not, proceed to item 2 in this table.

Item	Description
2	If local authentication is disabled on the system or if the administrative user's username is not configured in the *local* context, the system determines if a domain was received as part of the username. If there is a domain and it matches the name of a configured context or domain, the systems uses the AAA configuration within that context. If there is a domain and it does not match the name of a configured context or domain, Go to item 4 in this table. If there is no domain as part of the username, go to item 3 in this table.
3	If there was no domain specified in the username or the domain is not recognized, the system determines whether an *AAA Administrator Default Domain* is configured. If the default domain is configured and it matches a configured context, the AAA configuration within the *AAA Administrator Default Domain* context is used. If the default domain is not configured or does not match a configured context or domain, go to item 4 item this table.
4	If a domain was specified as part of the username but it did not match a configured context, or if a domain was not specified as part of the username, the system determines if the *AAA Administrator Last Resort context parameter* is configured. If a last resort context is configured and it matches a configured context, the AAA configuration within that context is used. If a last resort context is not configured or does not match a configured context or domain, the AAA configuration within the *local* context is used.

Figure 4. Context-level Administrative User AAA Context



Context Selection for Subscriber Sessions

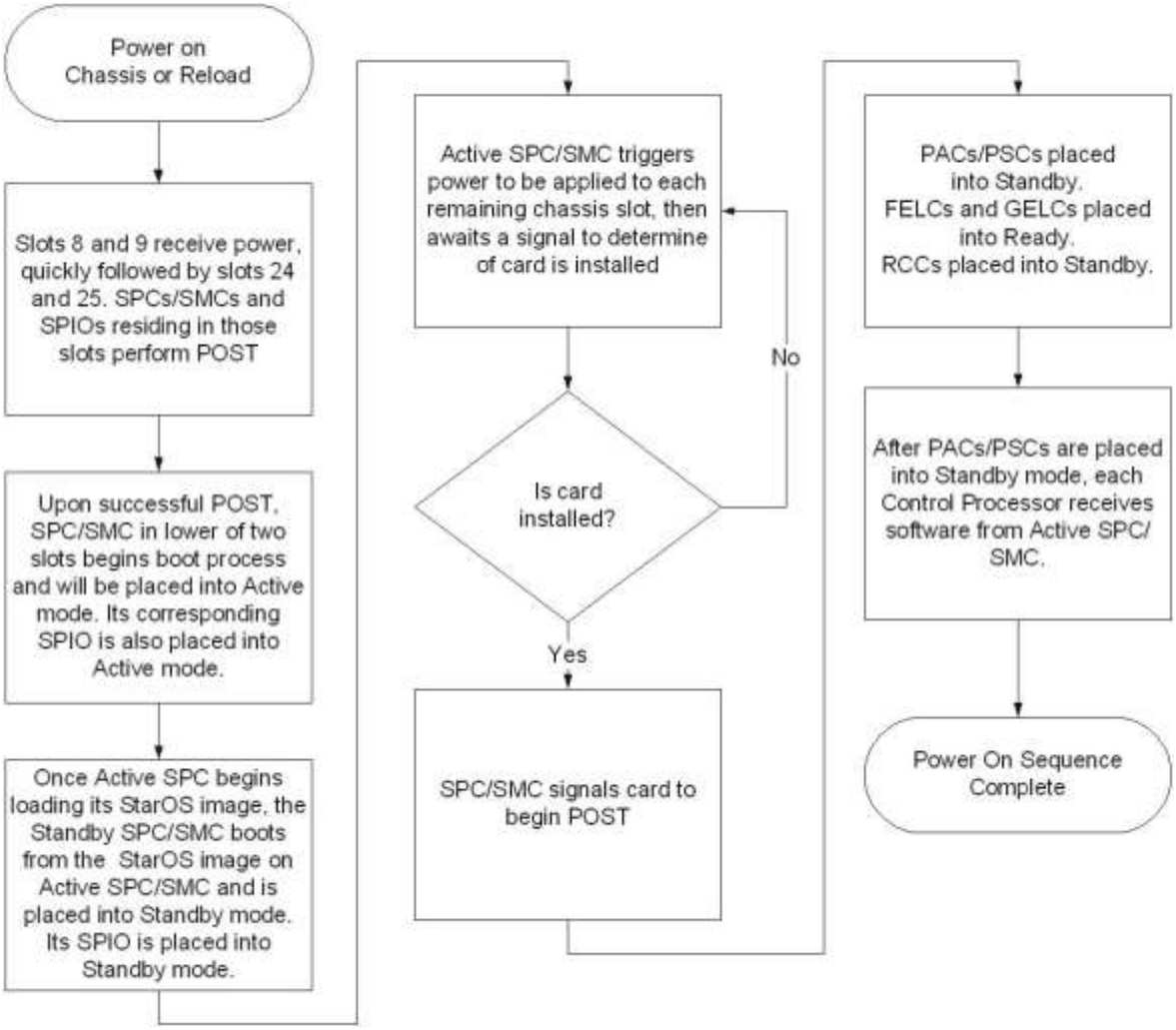
The context selection process for a subscriber session is more involved than that for the administrative users. Subscriber session context selection information for specific products is located in the Administration Guide for the individual product.

Understanding the System Boot Process

Refer to other sections of this guide for information and instructions on configuring the system. Part of the configuration process requires that you allocate hardware resources for processing and redundancy. Therefore, before you configure the system, it is important to understand the boot process which determines how the hardware components are brought on line.

The following flowchart shows each step in the startup process. For additional information about system configuration files, refer to the Understanding Configuration Files section.

Figure 5. Boot Process Flowchart



The following steps describe the system's boot process:

Step 1 When power is first applied to the chassis, or after a reboot, only the SPC/SMC slots (slots 8 and 9) receive power. Therefore, the SPCs/SMCs are the first cards to boot and their LEDs are the first to light up. After the system confirms that cards are located in slots 8 and 9, power is quickly applied to the SPIOs in slots 24 and 25.

Step 2 During the startup process, each card performs a series of power-on self tests (POSTs) to ensure that the hardware is operational.

Step 3 If the SPC/SMC in slot 8 successfully executes all POSTs, the card in slot 8 becomes the active SPC/SMC. The SPC/SMC in slot 9 becomes the standby card. If there is a problem with the SPC/SMC in slot 8, the card in slot 9 becomes the active SPC/SMC. Once the active and standby order is determined, the SPIO cards in slots 24 and 25 are placed into active and standby mode, as determined by the direct mapping of the active and standby SPCs/SMCs.

Step 4 The active SPC/SMC begins loading the operating system software image designated in the boot stack. The boot stack entries are contained in the boot.sys file that resides on the SPC/SMC CompactFlash. The standby SPC/SMC observes the active card startup. If the file on the active card is loads normally, the standby SPC/SMC boots from the active card image. If the active SPC/SMC experiences problems during this phase, the standby card loads its software image designated by its own boot stack entry in its boot.sys file and takes over control of the system as the active card.

Step 5 After the software image is loaded into SPC/SMC RAM, the active card determines whether other cards are installed in the chassis by applying power to the other chassis slots and signalling them. If the chassis slot contains an application or line card, power is left on to that slot. All empty slots are powered off.

> ***Important:*** If no SPCs/SMCs are installed, or if they are installed incorrectly, no other card installed in the system will boot.

Step 6 When power is applied to the PACs, PSCs and line cards installed in the system, they each perform their own series of POSTs.

Step 7 After successful POST, each of the PACs/PSCs enter the standby mode.

Installed line cards all remain in a ready mode until their corresponding PAC/PSC is made active via configuration. After the PAC/PSC is made active, the line card installed in the upper-rear chassis slot behind the card is also made active. The line card installed in the lower-rear chassis slot behind the card enters the standby mode.

Step 8 After entering the standby mode, each of the PAC/PSC control processors (CPs) communicate with the SPC/SMC to receive the appropriate code.

Step 9 Upon successful loading of the software image, the system loads a configuration file designated in the boot stack (boot.sys file). If this is the first time the system is powered on and there is no configuration file, the active SPC/SMC invokes the system's Quick Setup wizard. Use the Quick Setup wizard to configure basic system parameters for communication across the management network.

The wizard creates a configuration file, system.cfg, that you can use as a starting point for subsequent configurations. This allows you to configure the system automatically by applying the configuration file during any subsequent boot. For additional information about system configuration files, refer to the *Understanding Configuration Files* section.

Understanding Configuration Files

The system supports the use of a file or script to modify configurable parameters. When you use a file for offline system configuration, it reduces the time it takes to configure parameters on multiple systems.

A system configuration file is an ASCII text file that contains commands and configuration parameters. When you apply the configuration file, the system parses through the file line by line, testing the syntax and executing the command. If the syntax is incorrect, a message is displayed to the CLI and the system proceeds to the next command. Lines that begin with # are considered remarks and are ignored.

Important: Pipes (|), used with the **grep** and **more** keywords, can potentially cause errors in configuration file processing. Therefore, the system automatically ignores keywords with pipes during processing.

The commands and configuration data within the file are organized and formatted just as they would be if they were being entered at the CLI prompt. For example, if you wanted to create a context called source in the CLI, you would enter the following commands at their respective prompts:

```
[local]host_name# config

[local]host_name(config)# context source

[source]host_name(config-ctx)# end
```

To create a context called source using a configuration file, you would use a text editor to create a new file that consists of the following:

```
config

context source

end
```

There are several important things to consider when using configuration files:

- The system automatically applies a configuration file at the end of the boot process. After the system boots up for the first time, a configuration file that you have created and that is tailored to your network needs, can be applied. To make the system use your configuration file, modify the system's boot parameters according to the instructions located in *Software Management Operations.*
- In addition to being applied during the boot process, you can also apply configuration files manually at any time by executing the appropriate commands at the CLI prompt. Refer to the instructions in *Software Management Operations*.

  Important: When you apply a configuration file after the boot process, the file does not delete the configuration loaded as part of the boot process. Only those commands that are duplicated are overwritten.

- Configuration files can be stored in any of the following locations:
  - **CompactFlash™:** Installed on the SPC or SMC

- **PCMCIA Flash Card:** Installed in a slot on the SPC or SMC
- **Network Server:** Any workstation or server on the network that the system can access using the Trivial File Transfer Protocol (TFTP). This is recommended for large network deployments in which multiple systems require the same configuration

- Each time you save configuration changes you made during a CLI session, you can save those settings to a file which you can use as a configuration file.

Chapter 3
Getting Started

The system is shipped with no active configuration file. As a result, you must configure the software after the hardware is fully installed and the installation verified.

This chapter provides instructions for connecting to the console port and for creating the initial Local context management configuration. It includes the following sections:

Configuration

The first time power is applied to the system, the Switch Processor Card (SPC) or System Management Card (SMC) installed in chassis slot 8 automatically launches a Quick Setup Wizard on its console port.

The console port is located at the upper-rear of the chassis on the Switch Processor Input/Output (SPIO) Line Card installed in slot 24. The purpose of this wizard is to guide you through the initial configuration of the system.

You can choose not to use the wizard and perform the initial configuration by issuing commands to the command line interface (CLI).

The following sections describe how to configure the system.

Using the Quick Setup Wizard

The Quick Setup Wizard consists of three parts:

- Configuring a context-level security administrator and hostname
- Configuring the Ethernet interface for out-of-band (OOB) management on the SPIO installed in slot 24
- Configuring the system for remote CLI access via Telnet, Secure Shell (SSH), or File Transfer Protocol (FTP)

The following figure and table provides a flow diagram that shows the logic of the wizard and additional information and notes.

Figure 6. System Quick Setup Wizard Logic Diagram

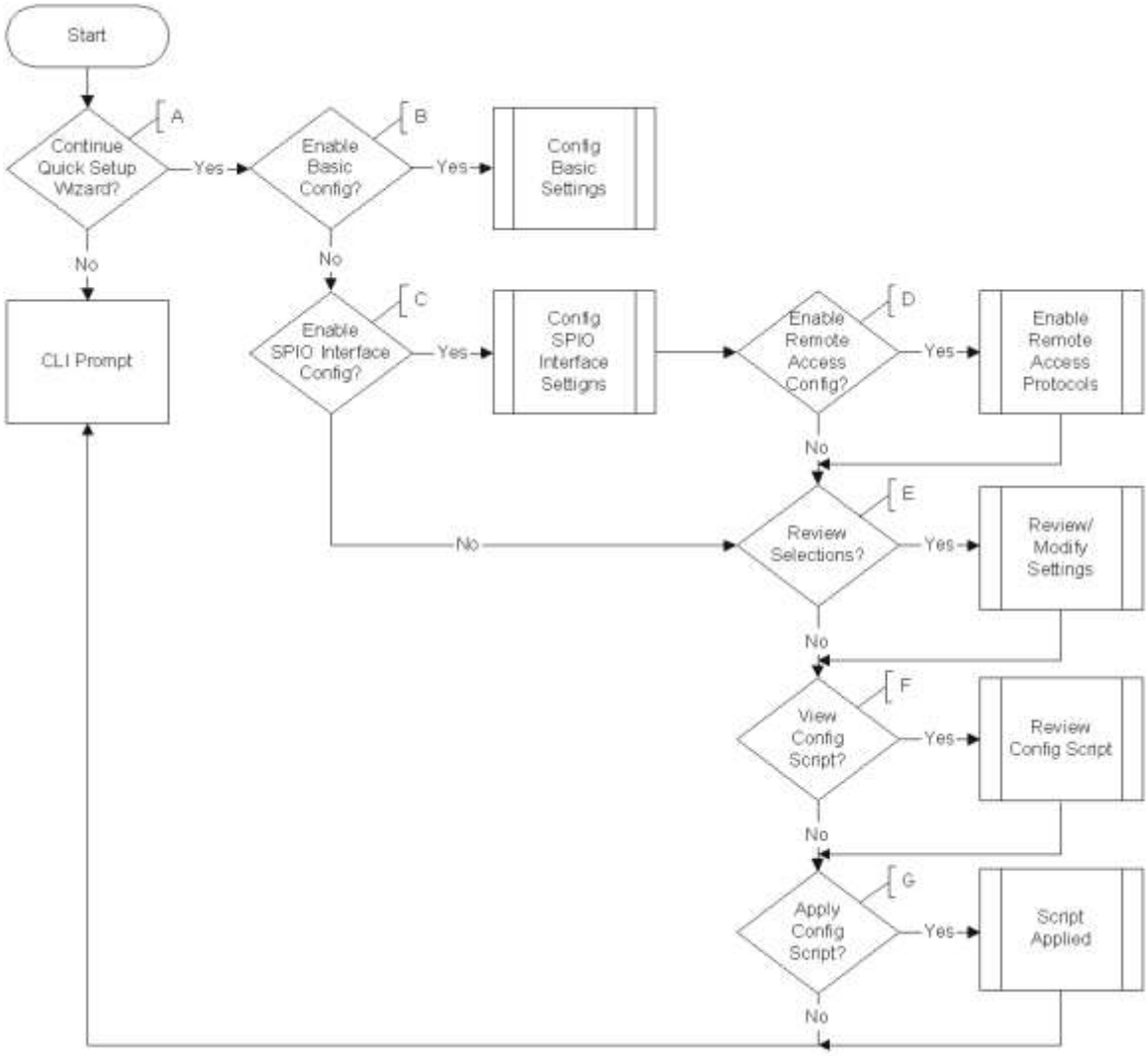


Table 7. System Quick Setup Wizard Logic Diagram Callout Descriptions

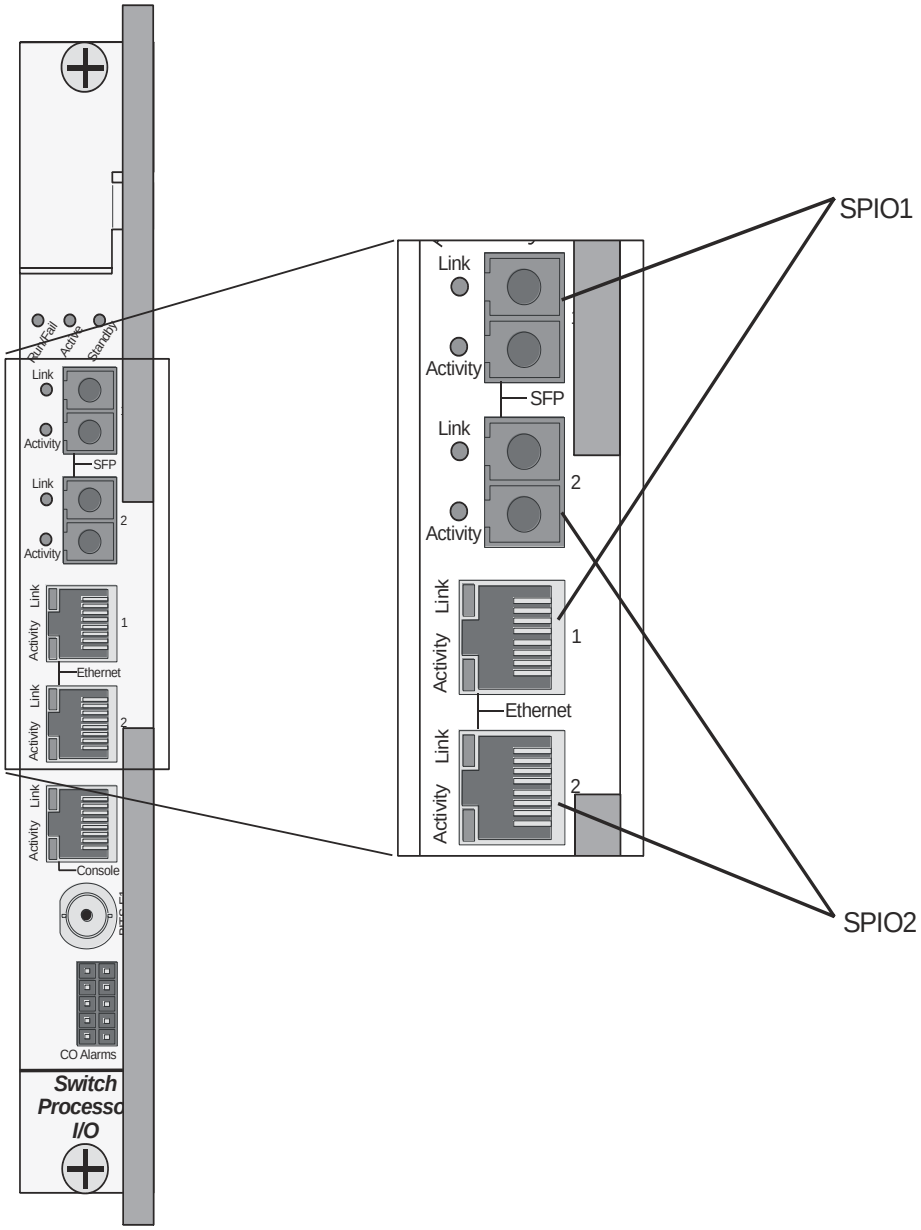
Callout	Description/Notes
A	Enter or exit the wizard. • Enter **no** at the prompt to automatically be directed to the system's CLI. Proceed to the Using the CLI for Initial Configuration section for instructions on performing an initial system configuration with the CLI. • Enter **setup** at the command prompt to re-invoke the wizard.
B	Configure an administrative username/password and the a hostname for the system. • The name of the default administrative user configured through the wizard is `admin`. • Administrative user names can be up to 32 alpha and/or numeric characters and are case sensitive. • Administrative user passwords can be up to 63 alpha and/or numeric characters and are case sensitive. • Configure a valid, non-null hostname. The hostname can consist of up to 63 alpha and/or numeric characters and is case sensitive.

C	Configure a single Switch Processor Input/Output (SPIO) out-of-band management interface for out-of-band system management.
	• Traffic on the management LAN is not transferred over the same media as user data and control signaling.
	• For security reasons, it is recommended that management functions be maintained on a separate network from user data and control signaling.
	• Depending on the medium being used to access the network, Ethernet or fiber:
	  • SPIO1 represents either the Ethernet 1 or SFP 1 interface on the SPIO, as shown in the figure below. *SPIO1* is the default.
	  • SPIO2 represents either the Ethernet 2 or SFP 2 interface on the SPIO.
	• Use the RJ-45 interfaces to connect the system to the management network with CAT3 or CAT5 Ethernet cable. Use the SFP interfaces to connect the system to the management network with 1000Base-SX optical fiber cable. The default is *rj-45*.
	• Configure an IP address, subnet mask, and gateway for the interface.
	• Instructions for configuring the second management interface on the SPIO can be found in *Configuring System Settings*.

D	Enable various remote access protocols for accessing the system.
	• Secure Shell (SSH) uses TCP port number 22 by default, if enabled.
	  • SSH V1 and/or V2 are supported.
	  • If SSH is enabled, you can also enable SSH File Transfer Protocol (SFTP) server functionality.
	• Telnet uses TCP port number 23 by default, if enabled.
	• The File Transfer Protocol (FTP) uses TCP port number 21 by default, if enabled.
	Important: For maximum security, it is recommended that you use only SSH v2.

Callout	Description/Notes
E	Review and/or modify the configuration of previous prompts. **1.** Enter the number of the prompt to be modified. **2.** Configure the parameter. **3.** *Optional*. Repeat *step 1* and *step 2* to modify additional settings. **4.** Enter “done” when you have completed all changes.
F	Review the configure script created by the wizard based on your inputs. An example of a created script is displayed in the example below. Variables are displayed in italics (`variable`).
G	Apply the configuration file to the system. Once applied, the parameter configuration is automatically saved to the system.cfg file stored on the primary SPC/SMC compact flash card.

Figure 7. SPIO Interfaces




```
config
 system hostname <hostname>
 context local
 administrator <admin_name> password <passwd>
 interface spio1
 ip address <ip_address> subnet
```

```
 #exit
 ip route 0.0.0.0 0.0.0.0 <gw_address> spio1
 ssh key <v1_key>
 ssh key <v2_rsa_key>
 ssh key <v2_dsa_key>
 server sshd
 subsystem sftp
 #exit
 no server telnetd
 server ftpd
 no server telnetd
 #exit
 port ethernet 24/1
 bind interface spio1 local
 no shutdown
 media rj45
 #exit
end
```

Important: Once configuration using the wizard is complete, proceed to instructions on configuring other system parameters.

Using the CLI for Initial Configuration

The initial configuration consists of the following:

- Configuring a context-level security administrator and hostname
- Configuring the Ethernet interface(s) on the SPIO that is installed behind the primary SPC/SMC
- Configuring the system for remote CLI access via Telnet, SSH, or FTP (secured or unsecured)

This section provides instructions for performing these tasks using the CLI.

Step 1 At the [local]host_name prompt enter:

```
configure
```

The following prompt appears:

```
[local]host_name(config)#
```

Step 2 Enter the context configuration mode by entering the following command:

```
context local
```

The *local* context is the system's management context. Contexts allow you to logically group services or interfaces. A single context can consist of multiple services and can be bound to multiple interfaces.

The following prompt appears:

```
[local]host_name(config-ctx)#
```

Step 3 Enter the following command to configure a context-level security administrator for the system:

```
administrator <name> { password <password> | encrypted password
<enc_password> } [ ftp ] [ no-cli ] [ timeout-absolute
<absolute_time> ] [ timeout-idle <idle_time> ]
```

Keyword/Variable	Description
<name>	Specifies the security administrator's name. The name can be between 1 and 32 alpha and/or numeric characters and is case sensitive.
password *<password>*	Specifies the password for the security administrator. The password can be between 1 and 63 alpha and/or numeric characters and is case sensitive.

Keyword/Variable	Description
encrypted password	Specifies the encrypted password for the security administrator. The keyword is only used by the system when you save configuration scripts. The system displays the encrypted keyword in the configuration file as a flag indicating that the variable following the keyword is the encrypted version of the plain text password. Only the encrypted password is saved as part of the configuration file.
ftp	Specifies that the security administrator is allowed to access the system with the File Transfer Protocol (FTP). This option is useful to upload files (configuration or software images) to the system's CompactFlash or PC-card Flash modules.
no-cli	Specifies that the security administrator cannot access the system's command line interface (CLI). ***Important:*** Use this keyword in conjunction with the ftp keyword to allow access to the system with FTP only.
timeout-absolute	Specifies the maximum amount of time that the operator can maintain a session with the system. The absolute_time is measured in seconds. Use any integer value between 0 and 300000000. The default absolute_time is 0. In the event that the absolute timeout value is reached, the operator session is automatically terminated.
timeout-idle	Specifies the maximum amount of time that an operator session can remain idle before being automatically terminated. The idle_time is measured in seconds. Use any integer value between 0 and 300000000. The default idle_time is 0.

Important: You must configure a context-level security administrator during the initial configuration. After you complete the initial configuration process and end the CLI session, if you have not configured a security administrator, CLI access will be locked.

Step 4 Enter the following command at the prompt to exit the context configuration mode:

```
exit
```

The following prompt appears:

```
[local]host_name(config)#
```

Step 5 Enter the following command to configure a hostname by which the system will be recognized on the network:

```
system hostname <host_name>
```

<host_name> is the name by which the system will be recognized on the network. The hostname can be up to 63 alpha and/or numeric characters and is case sensitive.

Step 6 Configure the network interfaces on the SPIO using the following instructions:

Step a Enter the context configuration mode by entering the following commands:

```
context local
```

The following prompt appears:

```
[local]host_name(config-ctx)#
```

Step b Enter the following command to specify a name for the interface:

```
interface <interface_name>
```

<interface_name> is the name of the interface. The interface name can be between 1 and 79 alpha and/or numeric characters and is case sensitive.

The following prompt appears as the system enters the Ethernet Interface Configuration mode:

```
[local]host_name(config-if-eth)#
```

Step c Configure an IP address for the interface configured in the previous step by entering the following command:

```
ip address <ipaddress> <subnetmask>
```

Variable	Description
ipaddress	Specifies the IP address for the interface.
subnetmask	Specifies the subnet mask for the interface.

Important: If you are executing this command to correct an address or subnet that was mis-configured with the Quick Setup Wizard, you must verify the default route and port binding configuration. Use *step 11* and *step 6* of this procedure. If there are issues, perform steps *7e* through *7k* to reconfigure the information.

Step d Enter the following command to exit the Ethernet interface configuration mode:

```
exit
```

The following prompt appears:

```
[local]host_name(config-ctx)#
```

Step e Configure a static route, if required, to point the system to a default gateway. Entering the following command:

```
ip route 0.0.0.0 0.0.0.0 <gw_address> <interface_name>
```

Variable	Description
gw_address	Specifies the IP address of the default gateway.
interface_name	Specifies the name of the interface that was configured in step 7b.

Step f Enter the following to exit from the context configuration mode:

```
exit
```

The following prompt appears:

```
[local]host_name(config)#
```

Step g Enter the Ethernet Port Configuration mode:

```
port ethernet <slot#>/<port#>
```

Variable	Description
slot#	The actual chassis slot in which the line card is installed. This could be either slot number 24 or 25.
port#	The physical port on the line card that will be used. For the SPIO, this will be either port 1 or 2. Port 1 represents the top most port (either RJ-45 or SFP).

The following prompt appears:

```
[local]host_name(config-port-<slot#/port#>)#
```

Step h Bind the port to the interface that you created in step 7b. Binding associates the port and all of its settings to the interface. Enter the following command:

```
bind interface <interface_name> local
no shutdown
```

<interface_name> is the name of the interface that you configured in *step 7b*.

Step i Specify which Ethernet media you are using. Enter the following:

```
media [ rj45 | sfp ]
```

The SPIO is equipped with dual RJ-45 and dual SFP interfaces. The RJ-45 interfaces connect the system to the management network with CAT3 or CAT5 Ethernet cable. The SFP interfaces connect the system to the management network with 1000Base-SX optical fiber cable.

Step j Configure the port speed, if needed, by entering the following command:

```
medium { auto | speed { 10 | 100 | 1000 } duplex { full | half } }
```

Keyword/Variable	Description
auto	Configures the system to auto detect the port speed. This is the default setting.
speed	Specifies the port speed for the port itself. When manually configuring the port speed, you must ensure that the network server configuration supports the speed and duplex configuration. The possible rates are: • 10 specifies 10 Mbps • 100 specifies 100 Mbps • 1000 specifies 1000 Mbps ***Important:*** Use 1000 Mbps only for the SFP ports on the Ethernet 1000 or SPIO Line Cards. In addition, if you manually configure the port speed, you must also configure the duplex mode.
duplex	If you manually configure the speed, you must also use this parameter to configure the duplex mode. You can implement either a full or half duplex mode. ***Important:*** Ethernet networking rules dictate that if a device whose interface is configured to auto-negotiate is communicating with a device that is manually configured to support full duplex mode, the first device negotiates with the manually configured speed of the second device, but only communicates in half duplex mode.

Step k Exit the Ethernet Interface Configuration mode by entering the command:

```
exit
```

The following prompt appears:

```
[local]host_name(config)#
```

Important: Refer below for instructions on configuring the SPIO management interface with a second IP address.

Configuring the System for Remote Access

Configure the system for remote access. An administrative user may access the system from a remote location over a local area network (LAN) or wide area network (WAN):

- Telnet
- Secure Shell (SSH)
- File Transfer Protocol (FTP) (secured or unsecured)
- Trivial File Transfer Protocol (TFTP)

Important: For maximum security, use SSH v2.

Step 1 Enter the context configuration mode by entering the following command:

```
context local
```

The following prompt appears:

```
[local]host_name(config-ctx)#
```

Step 2 Configure the system to allow Telnet access, if desired:

```
server telnetd
```

Step 3 Configure the system to allow SSH access, if desired:

```
ssh generate key [ type { v1-rsa | v2-rsa | v2-dsa } ]
```

Important: **`v2-rsa`** is the recommended key type.

```
server sshd
```

Step 4 Configure the system to allow FTP access, if desired, by entering the following command:

```
server ftpd
```

Step 5 Configure the system to allow TFTP access, if desired:

```
server tftpd
```

Step 6 Exit the context configuration mode by entering the following command:

```
exit
```

The following prompt appears:

```
[local]host_name(config)#
```

Step 7 Exit the configuration mode by entering the following command:

```
end
```

The following prompt appears:

```
[local]host_name#
```

Step 8 Verify the configuration by entering the following command:

```
show configuration
```

The CLI output should be similar to the sample output:

```
  context local
 interface <interface_name>
 ip address <ipaddress> <subnetmask>
 #exit
 subscriber default
 #exit
 administrator <admin_name> password <admin_password>
      server telnetd
      server ftpd
      ssh generate key
      server sshd
  #exit
```

```
port ethernet 24/1
   bind interface <interface_name> local
#exit
port ethernet 24/1
   no shutdown
#exit
snmp engine-id local 800007e580ed826c191ded2d3d
end
```

Step 9 Verify the configuration of the IP routes by entering the following command:

```
show ip route
```

The CLI output should be similar to the sample output:

```
"*" indicates the Best or Used route.
 Destination Nexthop Protocol Prec Cost Interface
*0.0.0.0/0 <ipaddress> static 1 0 spio1
*<network> 0.0.0.0 connected 0 0 spio1
```

Step 10 Verify the interface binding by entering the following command:

```
show ip interface name <interface_name>
```

<interface_name> is the name of the interface that was configured in *step 7b*.

The CLI output should be similar to the sample output:

```
Intf Name: spio1Intf Type: Broadcast
Description:
IP State: UP (Bound to 24/1 untagged, ifIndex 402718721)
IP Address: <ipaddress> Subnet Mask: <subnetmask>
Bcast Address: <bcastaddress> MTU: 1500
Resoln Type: ARP ARP timeout: 3600 secs
L3 monitor LC-port switchover: Disabled
```

```
                    Number of Secondary Addresses: 0
```

Step 11 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring the SPIO Management Interface with a Second IP Address

If necessary, you can configure a second IP address on the SPIO management interface.

Step 1 Enter the configuration mode by entering the following command at the prompt:

```
configure
```

The following prompt appears:

```
[local]host_name(config)#
```

Step 2 Enter the following to enter the context configuration mode:

```
context local
```

The following prompt appears:

```
[local]host-name(config-ctx)#
```

Step 3 Enter the interface slot number and port number by entering the following command:

```
[local]host_name(config-ctx)# 24/1
```

The following prompt appears:

```
[local]host_name(config-if-eth)#
```

Step 4 Enter the secondary IP address and subnet mask by entering the following command:

```
[local]host_name(config-if-eth)#ip address xxx.xxx.xxx.xxx
xxx.xxx.xxx.xxx secondary
```

Step 5 Exit the configuration mode by entering the following command:

```
[local_host]host_name(config-if-eth)#end
```

Step 6 Confirm the interface ip addresses by entering the following command:

```
[local_host]# show config context local
```

The CLI output should look similar to this example:

```
config
   context local
      interface <interface_name>
         ip address <ipaddress> <subnetmask>
         ip address <ipaddress> <subnetmask> secondary
         #exit
```

Step 7 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Chapter 4
Configuring System Settings

This chapter provides instructions for configuring system options such as:

- Configuring a Second Management Interface
- Configuring System Timing
- Configuring Transmit Timing Source
- Enabling CLI Timestamping
- Configuring System Administrative Users
- Configuring Virtual MAC Addresses
- Configuring PAC/PSC and Line Card Availability
- Configuring Line Card and SPIO Port Redundancy
- Configuring Link Aggregation

Important: The contents of this chapter assume that the procedures to initially configure the system in the *Getting Started* chapter have been completed.

Important: The commands used in the configuration examples in this section are the most common or most likely-used commands and/or keyword options. In many cases, other optional commands and/or keyword options are available. Refer to the *Command Line Interface Reference* for complete information.

Configuring a Second Management Interface

Refer to *Getting Started* for instructions on configuring a system management interface on the SPIO. This section provides instructions for configuring a second management interface.

Use the following example to configure a second management interface:

```
configure
   context local
      interface <interface_name>
         ip address <ipaddress subnetmask>
         exit
      ip route 0.0.0.0 0.0.0.0 <gw_address interface_name>
      exit
   port ethernet <slot#/port#>
      bind interface <interface_name>local
      no shutdown
      media [ rj45 | sfp ]
      end
```

Notes:

- For **port ethernet** *slot#*, use the actual chassis slot in which the SPIO is installed. This could be either slot number 24 or 25.
- For **port ethernet** *port#*, use the physical port on the SPIO card that will be used. For the SPIO, this is either port 1 or 2. Port 1 represents the top-most port (either RJ-45 or SFP).
- The SPIO is equipped with dual RJ-45 and dual SFP interfaces. The RJ-45 interfaces connect the system to the management network with CAT3 or CAT5 Ethernet cable. The SFP interfaces connect the system to the management network with 1000Base-SX optical fiber cable.
- *Option:* In the Ethernet Port configuration mode, configure the port speed, if needed, by entering the medium command. Refer to the *Command Line Interface Reference* for a complete explanation of this command.

Verifying and Saving Your Interface and Port Configuration

Step 1 Verify that your interface configuration settings are correct by entering the following command:

show ip interface

The output from this command should look similar that shown below. In this example an interface named mgmt2 was configured in the local context.

```
Intf Name: mgmt2

IP State: UP (Bound to 24/2)

IP Address: 192.168.100.3 Subnet Mask: 255.255.255.0

Bcast Address: 192.168.100.255 MTU: 1500

Resoln Type: ARP ARP timeout: 3600

Number of Secondary Addresses: 0
```

Step 2 Verify that the port configuration settings are correct by entering the following command:

show configuration port *<slot#/port#>*

slot# is the chassis slot number of the line card where the physical port resides. slot# is either 24 or 25. *port#* is the number of the port (either 1 or 2).

This command produces an output similar to the one shown below that displays the configuration of port 2 of the SPIO installed in chassis slot 24. In this example, the port is bound to an interface called mgmt2.

```
config

 port ethernet 24/2

 description management2

 no shutdown

 bind interface mgmt2 local

 #exit

end
```

Step 3 Save your configuration as described in the *Saving Your Configuration* chapter.

Configuring System Timing

The system is equipped with a system clock that supplies the timestamp for statistic counters, accounting records, logging, and event notification. After the initial configuration of the system clock, you can configure the system to communicate with one or more Network Time Protocol (NTP) server(s) on the network to ensure that the clock is always accurate.

In the event of a power outage, the clock is maintained with an accuracy of +/- one minute per month for up to 10 years. This ensures that when power is restored, the system is ready to process sessions and generate accounting, log, and event data with accurate timestamps.

In addition to configuring the timing source, you must configure the system's time zone.

Setting the System Clock and Time Zone

Use the following command example to configure the system clock and time zone:

```
clock set <date:time>

configure

   clock timezone <timezone> [ local ]

   end
```

Notes:

- Refer to the **clock timezone** command in the *Command Line Interface Reference* for a complete list of supported time zones.
- The optional **local** keyword indicates that the time zone specified is the local timezone.
- Daylight Savings Time is automatically adjusted for time zones supporting it.

Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying and Saving Your Clock and Time Zone Configuration

Enter the following command to verify that you configured the time and time zone correctly:

```
show clock
```

The output displays the date, time, and time zone you configured.

Configuring Network Time Protocol Support

This section provides information and instructions for configuring the system to enable the use of the Network Time Protocol (NTP).

Important: Configure the system clock and time zone prior to implementing NTP support. This greatly reduces the time period that must be corrected by the NTP server.

Configuring Network Time Protocol Support

This section provides information and instructions for configuring the system to enable the use of the Network Time Protocol (NTP).

Important: Configure the system clock and time zone prior to implementing NTP support. This greatly reduces the time period that must be corrected by the NTP server.

Use the following example to configure the necessary NTP association parameters:

```
configure
   ntp
      enable [ <context_name> ]
     server <ip_address>
      end
```

Notes:

- *context_name* is the name of a configured context other than *local*. Use this option to configure the system to run NTP in a specified context. By default, NTP runs in the local *context*. This is the recommended configuration.
- A number of options exist for the **server** command. Refer to the *NTP Configuration Mode Commands* chapter in the *Command Line Interface Reference* for more information.

Important: Configure the system with at least two NTP servers. We recommend four servers.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying the NTP Configuration

Step 1 Verify the NTP configuration is correct. Enter the following command at the Exec mode prompt:

```
show ntp associations
```

The output displays information about all NTP servers. See the output below for an example deploying two NTP servers.

+	-----Peer Selection:	( ) - Rejected/No response
|		(X) - False Tick
|		(.) - Excess
|		(-) - Outlier
|		(+) - Candidate
|		(#) - Selected
|		(*) - System Peer
|		(o) - PPS Peer
v		

remote	refid	st	t	when	poll	reach	delay	offset	jitter
==========									
+192.68.11.1	192.68.11.55	3	-	677	1024	377	0.800	1.330	1.111
#11.11.1.10	11.11.1.55	3	-	677	1024	377	0.725	-3.134	0.112

The following table describes the show ntp associations output table.

Column Title	Description
remote	Lists the current NTP servers. One of these characters precedes each IP address to show the server's current condition: • ( ) Rejected/No response • X False tick • . Excess • - Outlier • + Candidate • # Selected • * System peer • (o) PPS peer
refid	Shows the last reported NTP reference to which the server is synchronizing.
st	Shows the NTP server stratum level.
t	Shows the communication type: broadcast, multicast, etc.
when	Shows the number of seconds since the last contact.

Column Title	Description
poll	Shows the polling interval between the system and the NTP server.
reach	Shows the octal value of the reachability shift register indicating which responses were received for the previous 8 polls to this NTP server.
delay	Shows the round-trip delay (in milliseconds) for messages exchanged between the system and the NTP server.
offset	Shows the number of milliseconds by which the system clock must be adjusted to synchronize it with the NTP server.
jitter	Shows the jitter in milliseconds between the system and the NTP server.

Configuring Transmit Timing Source

This feature is only for application services that use SDH or SONET over the Optical or Channelized line cards.

In general, the SPIO automatically provides clocking based on the system clock. However, some application services that use SDH or SONET require greater clocking precision to ensure synchronous transmission. The timing source options include Building Integrated Timing Supply (BITS-timing) and line-timing. BITS-timing uses Stratum 3-compliant BITS modules resident on the SPIOs. The line-timing recovers the receive timing from an external clock through a specified port on an Optical or Channelized line card. The timing is then distributed via the SPIO to all line cards in the chassis.

Important: To use BITS-timing, the SPIO version must include the BITS BNC timing interface or the BITS 3-pin timing interface. For additional interface information, refer to the *Product Overview*.

You can enable and configure up to four timing sources: two BITS-timing and two line-timing sources. Having more than one timing source ensures back-up. If BITS-timing is enabled, it always takes priority over line-timing to provide the clocking.

Configure BITS as the Timing Source

Use the following example to configure BITS as the timing source:

```
configure
   port bits <slot#/port#>
      mode <e1/t1> framing <type>
      no shutdown
      end
```

Save the configuration according to the steps in the *Verifying and Saving Your Configuration* chapter.

Configure Line-timing as the Timing Source

Use the following example to configure line-timing as the timing source:

```
configure
   port atm <slot#/port#>
      line-timing
      no shutdown
      exit
   port bits <slot#/port#>
      recover line1 <linecard slot #>
      shutdown
      end
```

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configure Both BITS and Line as Timing Sources

Use the following example to configure both BITS and line-timing as the timing sources:

```
configure
   card <CLC slot#>
      framing <mode>
      exit
   port atm <OLC slot#/port#>
      line-timing
      no shutdown
      exit
   port channelized <CLC slot#/port#>
      line-timing
      no shutdown
```

```
        exit
    port bits <slot#/port#>
        recover line1 <LC slot#/port#>
        recover line2 <LC slot#/port#>
        no shutdown
        end
```

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Confirming the Timing Source

Use the **show timing** command, documented in the *Exec Mode Commands* chapter of the *Command Line Interface Reference*, to confirm the timing source has been configured correctly.

Enabling CLI Timestamping

To display a timestamp (date and time) for every command that is executed on the CLI, enter the following command at the root prompt for the Exec mode:

```
timestamps
```

Immediately after you execute the command, the date and time appear.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring System Administrative Users

The *Getting Started* chapter describes how to configure a context-level security administrator for the system.

This section provides instructions for configuring additional administrative users of each of the following privileges:

- security administrator
- administrator
- operator
- inspector

Instructions are categorized according to the type of administrative user: context-level, or local-user.

Important: For information on the differences between these user privileges and types, refer to the *Getting Started* chapter.

If your deployment does not require the configuration of additional administrative users, proceed to the *Configuring* PAC/*PSC and Line Card Availability* section.

Configuring Context-level Administrative Users

This section contains information and instructions for configuring context-level administrative user types.

Configuring Context-level Security Administrators

Use the example below to configure additional security administrators:

```
configure
   context local
      administrator <name> { password <pwrd> | encrypted password <pwrd>
}
      end
```

Notes:

- Additional keyword options are available that, for example, identify the administrator when active or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **administrator** command.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Context-level Administrators

Use the example below to configure context-level administrators:

```
configure
   context local
      config-administrator <name> { password <pwrd> | encrypted password <pwrd>
}
      end
```

Notes:

- Additional keyword options are available that, for example, identify the administrator when active or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **config-administrator** command.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Context-level Operators

Use the example below to configure context-level operators:

```
configure
   context local
      operator <name> { password <pwrd> | encrypted password <pwrd> }
      end
```

Notes:

- Additional keyword options are available that for example, identify the administrator when active or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **operator** command.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Context-level Inspectors

Use the example below to configure context-level inspectors:

```
configure
   context local
      inspector <name> { password <pwrd> | encrypted password <pwrd> }
      end
```

Notes:

- Additional keyword options are available that, for example, identify the administrator when active or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **inspector** command.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying Context-level Administrative User Configuration

Verify that the configuration was successful by entering the following command:

show configuration context local

This command displays all of the configuration parameters you modified within the Local context during this session. The following displays a command output sample. In this example, a security administrator named *testadmin* was configured.

```
config
 context local
 interface mgmt1
 ip address 192.168.1.10 255.255.255.0
 #exit
 subscriber default
 #exit
 administrator testadmin encrypted password fd01268373c5da85
    inspector testinspector encrypted password 148661a0bb12cd59
#exit
 port ethernet 24/1
 bind interface mgmt1 local
 #exit
end
```

Configuring Local-User Administrative Users

Use the example below to configure local-user administrative users:

configure

local-user username <*name*>

end

Notes:

- Additional keyword options are available that, for example, identify the administrator when active or place time thresholds on the administrator. Refer to the *Command Line Interface Referenc*e for more information about the **local-user username** command.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying Local-User Configuration

Verify that the configuration was successful by entering the following command:

show local-user verbose

This command displays information on configured local-user administrative users. The output below displays a sample of this command's output. In this example, a local-user named *SAUser* was configured.

```
Username: SAUser

 Auth Level: secadmin

 Last Login: Never

 Login Failures: 0

 Password Expired: Yes

 Locked: No

 Suspended: No

 Lockout on Pw Aging: Yes

 Lockout on Login Fail: Yes
```

Configuring Virtual MAC Addresses

When you enable virtual MAC addressing, a single block of 256 addresses is added to the system configuration. The MAC addresses assigned and stored in the IDEEPROM on Ethernet Line Cards are disregarded; MAC addresses for all ports on all Ethernet Line Cards are assigned from the specified block of virtual MAC addresses. This does not affect the MAC addresses on SPIO cards.

As in normal MAC address assignments, the corresponding ports on the upper and lower line cards have the same assigned MAC address. When you enable virtual MAC addressing, these addresses are all assigned from the specified block of 256 addresses.

If you enable virtual MAC addressing and remove a line card from the system, MAC addresses do not have to be reassigned because the MAC addresses in use do not belong to any line card. Therefore, if a line card is removed from the system, there is no possibility that any port on a linecard in the system is using any of the MAC addresses that belong to the removed line card.

Use the following example to configure virtual MAC addressing:

```
configure
   port mac-address virtual-base-address <MAC_Address>
   end
```

Notes:

- *MAC_Address is* the first address of a block of 256 MAC addresses. The system has reserved 65536 MAC addresses (00:05:47:FF:00:00 - 00:05:47:FF:FF:FF) for use by customers. This range allows you to create 256 address blocks each containing 256 MAC addresses (e.g., 00:05:47:FF:00:00, 00:05:47:FF:01:00, 00:05:47:FF:02:00, 00:05:47:FF:03:00, 00:05:47:FF:04:00, etc.).

Caution: This configuration requires a valid block of unique MAC addresses that are not used anywhere else. The use of non-unique MAC addresses can degrade and impair the operation of your network.

Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying Virtual MAC Address Configuration

Verify port information by entering the following command

```
show port info <slot#/port#>
```

slot# is the chassis slot number of the line card on which the physical port resides. *port#* is the physical port on the line card.

The output of this command should be similar to that shown in the example below.

```
Port: 36/8

 Port Type : 10/100 Ethernet
```

```
Description : (None Set)

Controlled By Card : 4 (Packet Accelerator Card)

Redundancy Mode : Port Mode

Redundant With : 20/8

Preferred Port : Non-Revertive

Physical ifIndex : 604504064

Administrative State : Enabled

Configured Duplex : Auto

Configured Speed : Auto

MAC Address : 00-05-47-02-04-3F

Link State : Up

Link Duplex : Full

Link Speed : 100 Mb

Logical ifIndex : 604504065

Operational State : Down, Standby
```

Configuring Packet Processing and Line Card Availability

As discussed in the *Understanding the System Boot Process* section of *Understanding System Operation and Configuration*, when the system boots up, all installed PACs/PSCs/PSC2s are placed into standby mode. You must activate some of these cards in order to configure and use them for session processing. Others may remain in standby mode to serve as redundant components.

When you activate an application card, the line card behind it shows up as attached and in a Ready state. Only when you bind a logical interface to one of the ports of the line card pair will the line cards assume an active and standby state.

This section provides instructions for activating PACs/PSCs/PSC2s and specifying their redundancy.

Important: Refer to the *Product Overview Guide* for information about system hardware configurations and redundancy.

Enter the following command to check the application card's operational status:

```
show card table
```

This command lists the PACs/PSCs/PSC2s and RCCs installed in the system by their slot number, their operational status, whether or not the card is a single point of failure (SPOF), and its attachment to a line card.

Important: For an ASR 5000, the output of this command would have indicated "PSC" or "PSC2" instead of "PAC" (ST16).

Check the line card operational status by entering the following command:

```
show linecard table
```

This command lists the line cards installed in the system by their slot number, their operational status, whether or not the card is a single point of failure (SPOF), and its attachment to a PAC/PSC/PSC2 or SPC/SMC.

Use the following example to configure PAC/PSC/PSC2 and line card availability:

```
configure
   card <slot_#>
      mode active { pac | psc }
      exit
   card-standby-priority <slot#_p1 slot#_p2 ... slot#_pn>
   end
```

Notes:

- When activating cards, remember to keep at least one card in standby mode for redundancy.
- Repeat for every other PAC/PSC in the chassis that you wish to activate.

- The **card-standby-priority** specifies the order in which the system will use standby PACs/PSCs as redundant components.
  - By default, the system uses the standby PAC, PSC or PSC2 in the highest-numbered slot (slot 16) as the first card to use for redundancy. This step is required if there are processing cards installed in the system that are in standby mode, and you want to configure the system to use an order other than the default.
  - *slot#_p1* is the chassis slot number of the standby PAC/PSC/PSC2 that you want to use first as a redundant component. *slot#_p2* is the chassis slot number of the standby processing card that you want to use second as a redundant component. *slot#_pn* is the chassis slot number of the standby PAC/PSC that you want to use as the last redundant component.

Example

A system has three PACs, PSCs or PSC2s that are in standby mode. They are installed in chassis slots 14, 15, and 16. If an active processing card fails, and you want the PAC or PSC in slot 15 to replace the failed PAC/PSC followed by the PAC/PSC in slot 14, enter the following command:

```
card-standby-priority 15 14
```

In the unlikely event that the PACs/PSCs in chassis slots 15 and 14 are unavailable, the system automatically uses the remaining standby PAC/PSC in slot 16 for redundancy.
Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying Packet Processing and Line Card Configurations

Verify that the configuration was successful. Depending on the type of card(s) you activated, enter either or both of the following commands:

```
show card table
```

```
show linecard table
```

Any card that you made active should now have an operational status of *Active*.

Configuring Line Card and SPIO Port Redundancy

Port redundancy for line cards and SPIOs provides an added level of redundancy that minimizes the impact of network failures that occur external to the system. Examples include switch or router port failures, disconnected or cut cables, or other external faults that cause a link down error.

Caution: To ensure that system line card and port-level redundancy mechanisms function properly, disable the Spanning Tree protocol on devices connected directly to any system port. Failure to turn off the Spanning Tree protocol may result in failures in the redundancy mechanisms or service outage.

By default, the system provides port-level redundancy when a failure occurs, or you issue the **`port switch to`** command. In this mode, the ports on active and standby line cards (for example, 17/1 and 33/1 have the same MAC address), but since only one of these ports may be active at any one time there are no conflicts. This eliminates the need to transfer MAC addresses and send gratuitous ARPs in port failover situations. Instead, for Ethernet ports, three Ethernet broadcast packets containing the source MAC address are sent so that the external network equipment (switch, bridge, or other device) can re-learn the information after the topology change. However, if an line card removal is detected, the system sends out gratuitous ARPs to the network because of the MAC address change that occurred on the specific port.

With port redundancy, if a failover occurs, only the specific port(s) become active. For example; if port 17/1 fails, then port 33/1 becomes active, while all other active ports on the line card in slot 17 remain in the same active state. In port failover situations, use the **`show port table`** or **`show linecard table`** commands to check that ports are active on both cards and that both cards are active.

Take care when administratively disabling a port that is one of a redundant pair. A redundant pair comprises both the active and standby ports—for example 17/1 and 33/1. If 17/1 is active, administratively disabling 17/1 through the CLI does not make 33/1 active. It disables both 17/1 and 33/1 because an action on one port has the same effect on both. Refer to *Enabling Line Card* and *SPIO Redundancy* below and *Creating and Configuring Ethernet Interfaces and Ports* in the *System Element Configuration Procedures* chapter.

If card-level redundancy is initiated, there is no port-level redundancy in a line card or SPIO failover. The standby line card or SPIO becomes active and all ports on that card become active. With line cards, the system automatically copies all the MAC addresses and configuration parameters used by the failed line card to its redundant counterpart. The ports on SPIOs keep their original MAC addresses, and the system automatically copies the failed SPIO's configuration parameters to its redundant counterpart. The PAC/PSC automatically re-routes to its redundant line card.

With the SPIO cards, any time there is a port or card switch gratuitous ARPs are sent.

Important: Be aware that in the case of a system with only one SPC/SMC and two SPIO cards, both SPIOs come up online. Automatic switching of Ethernet ports does not occur in this scenario, but you can initiate card and port switching by using the **`card spio switch to`** and **`port switch to`** commands.

Port redundancy can be configured to be revertive or non-revertive. With revertive redundancy service is returned to the original port when service is restored.

This feature requires specific network topologies to work properly. The network must have redundant switching components or other devices that the system is connected to. The following diagrams show examples of a redundant switching topologies and how the system reacts to various external network device scenarios.

Figure 8. Network Topology Example Using Line Card Port Redundancy

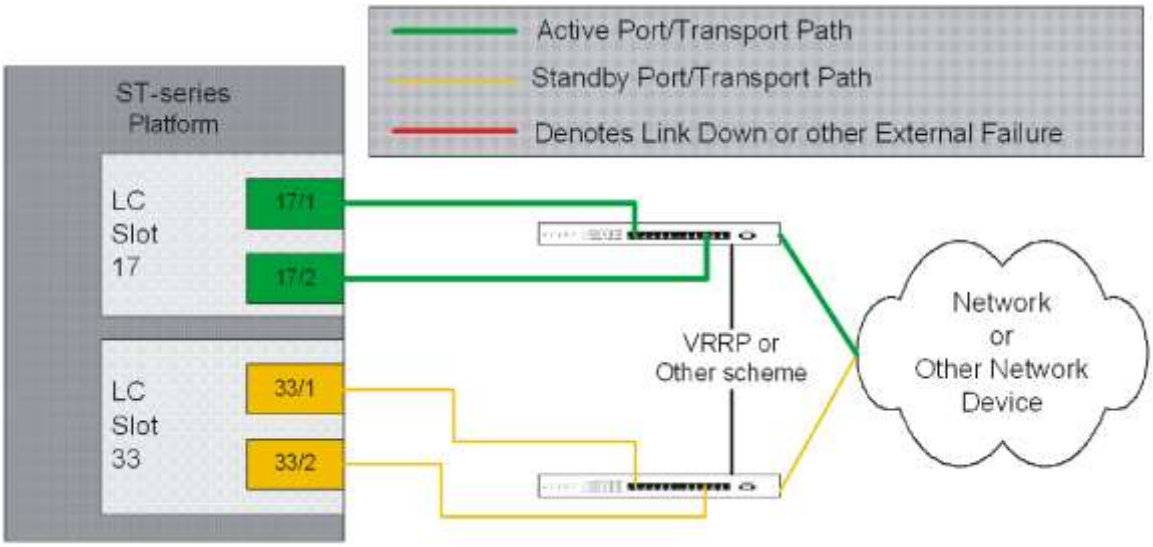
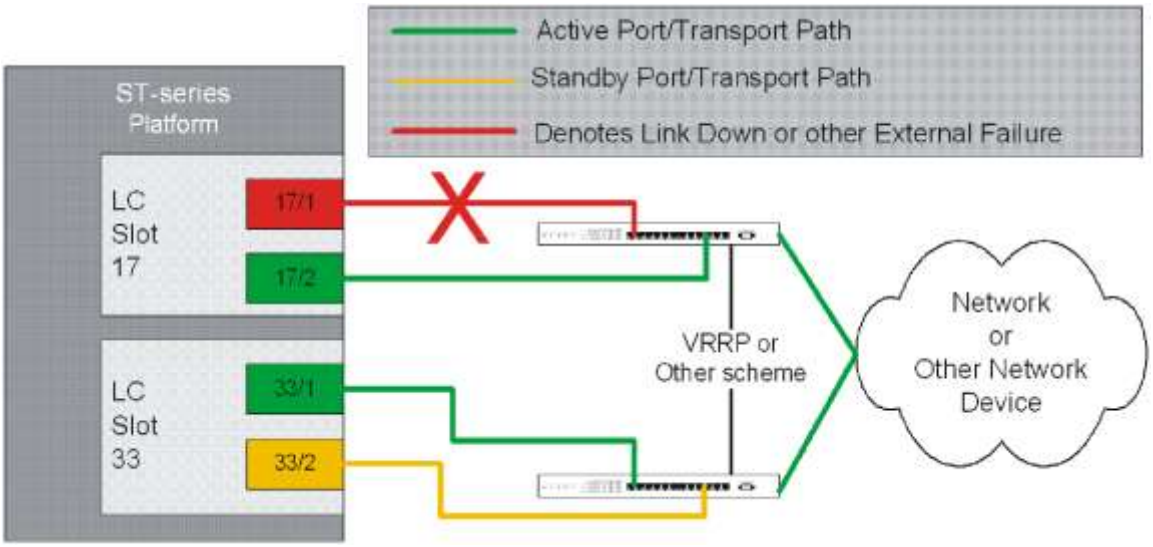
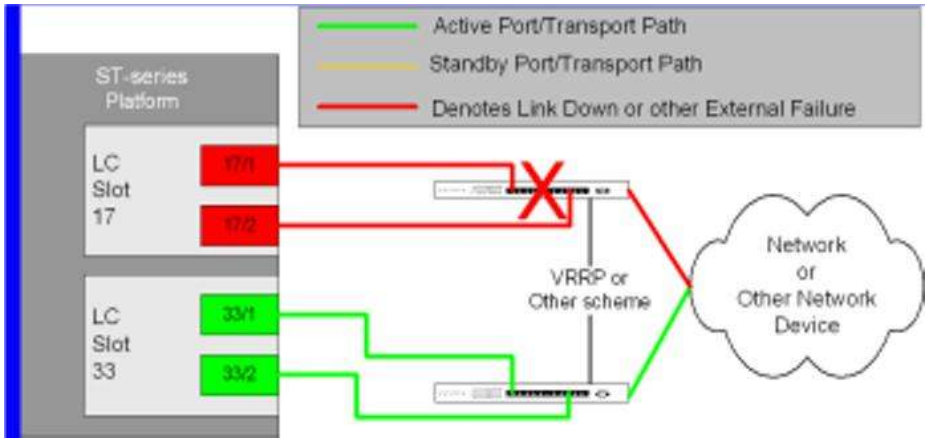


Figure 9. PortRedundancyFailover in Cable Defect Scenario



In the example above, an Ethernet cable is cut or unplugged, causing the link to go down. When this event occurs, the system, with port-mode redundancy enabled, recognizes the link down state and makes port 33/1 the active port. The switching devices, using some port redundancy scheme, recognizes the failure and enables the port on the secondary switch that the line card in slot 33 is connected to, allowing it to redirect and transport data.

Figure 10. Port Redundancy Failover in External Network Device Failure Scenario



In the example above, a switch failure causes a link down state on all ports connected to that switch. This failure causes all redundant ports on the line card in slot 33 to move into the active state and utilize the redundant switch.

Enabling Line Card and SPIO Port Redundancy

Use the following example to enable port redundancy:

```
configure
   card <slot_#>
      redundancy { card-mode | mixed-mode | port-mode }
      end
```

Notes:

- The **card-mode** keyword indicates that no port redundancy is used. The system provides card-level redundancy, which is triggered by an internal failure. The **port-mode** keyword, available for Ethernet and SPIO line cards, indicates that port redundancy will be enabled. This is the default redundancy mode.

Important: You do not need to use this configuration for each line card or SPIO. The system intuitively understands that if the command is entered for an active line card, the standby line will operate in the same mode. For example, if you enter the command for the line card in slot 17, it automatically places the line card in Slot 33 into port redundant operation.

Important: If you network-boot a dual-SPC/SMC chassis with SPIO port redundancy enabled, you should have CFE1.1.0 or greater in flash on both SPCs/SMCs. Otherwise, you risk having a standby SPC/SMC that can't boot from the network in certain circumstances. You can use any version of the CFE with SPIO port redundancy if the SPCs/SMCs boot from a local file system (/flash, /pcmcia1, or /pcmcia2).

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying Line Card and SPIO Port Redundancy

View the configuration of the card by entering the following command:

show configuration card <*slot_#*>

slot_# is the chassis slot number where the line card or SPIO you want to configure is installed.

The following is a sample of output for an line card in slot 17 and a SPIO in slot 24 that both have redundancy enabled.

```
[local]host_name# show config card 17
config
 card 17
 redundancy port-mode
 #exit
end
[local]host_name# show config card 24
config
 card 24
 redundancy port-mode
 #exit
end
```

Configuring Line Card and SPIO Port Redundancy Auto-Recovery

When port redundancy is enabled at the card level, you can configure a port auto-recovery feature. When a port failure occurs and the preferred port is returned to service (link is up), control is automatically returned to that port. By default, ports are in a non-revertive state, meaning that no ports are preferred, requiring a manual port switch to return use to the original port.

Important: This feature is applied on a per port basis, allowing you to configure specific ports to be used on individual line cards or SPIOs. For example, you could configure ports 1 through 4 as preferred on the line card in slot 17, and configure ports 5 through 8 as the preferred ports on the line card in slot 33. On a SPIO, you could configure port 1 as preferred on the SPIO in slot 24 and configure port 2 as preferred on the SPIO in slot 25. In this scenario, both line cards or SPIOs would be in an active state while providing line card and port redundancy for the other.

Use the following example to configure a preferred port for revertive, automatic return to service when a problem has cleared:

```
configure
   port ethernet <slot#/port#>
      preferred slot <slot#>
      end
```

Notes

- If you do not specify a preference, redundancy is non-revertive. If you do specify a preference, redundancy is revertive to the specified card.
- Repeat for each additional port that you want to make preferred.

Caution: A preference cannot be configured in normal redundancy mode. Attempting to do so will produce an error message from the cli command.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying Line Card and SPIO Port Redundancy Auto-Recovery

Verify port information by entering the following command

show port info *<slot#/port#>*

slot# is the chassis slot number of the line card on which the physical port resides.

port# is the physical port on the line card.

The following shows a sample output of this command for port 1 on the LC in slot 17:

```
[local]host_name# show port info 17/1
Port: 17/1
 Port Type : 10/100 Ethernet
 Description : (None Set)
 Controlled By Card : 1 (Packet Accelerator Card)
 Redundancy Mode : Port Mode
 Redundant With : 33/1
 Preferred Port : Revertive to port 17/1
 Physical Index : 285278208
 Administrative State : Disabled
```

```
Configured Duplex : Auto

Configured Speed : Auto

MAC Address : 00-05-47-01-11-00

Link State : Up

Link Duplex : Unknown

Link Speed : Unknown

Logical ifIndex : 285278209

Operational State : Down, Active
```

Configuring Link Aggregation

The four-port Quad Gigabit Ethernet line card (QGLC) developed for use in the ASR 5000 chassis supports link aggregation as defined in IEEE 802.3ad. No other proprietary or non-standard modes are supported.

Link aggregation (also called trunking or bonding) provides higher total bandwidth and better fault tolerance by combining up to four parallel network links between devices as a single link. A large file is guaranteed to be sent over one of the links, which removes the need to address out-of-order packets.

Important: The aggregated ports must be on the same QGLC redundant pair. Link aggregation does not work across line card slots. In the event of a failure of one or more of the member physical ports, the remaining ports continue to be aggregated.

Important: An aggregation group can consist of from one to four ports. A port can only be in one aggregation group; for example, Port 3 can be in Group A linked to Switch 1, but it cannot simultaneously be in Group B linked to Switch 2.

Requirements

Observe the following requirements:

- Assure that links between the two systems are full duplex and at the same speed.
- Set the port medium configuration to auto or full duplex and maximum speed.
- Certain physical port configuration changes, such as the MAC address or SRP, are prohibited on any interface participating in link aggregation

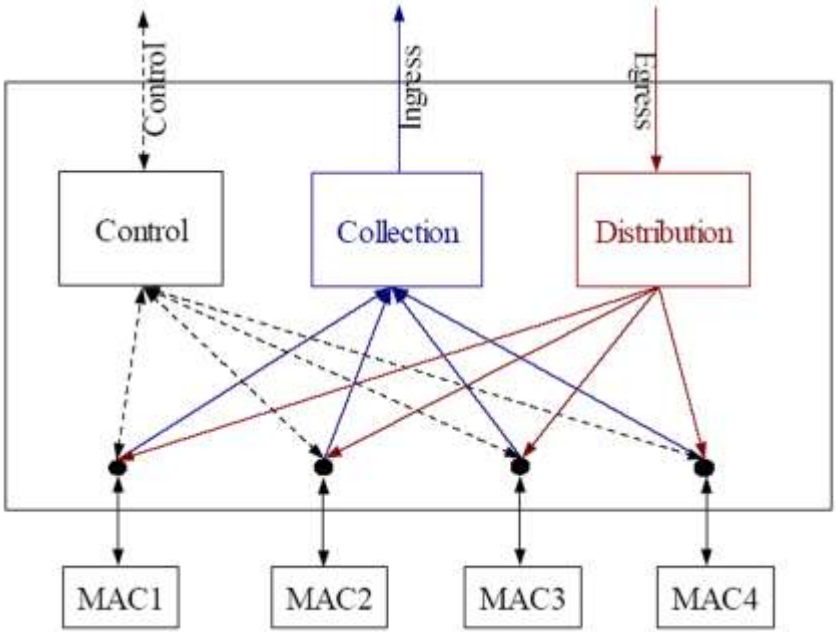
There is more on configuring ports and port redundancy in “Configuring Line Card and SPIO Port Redundancy.”

Operation

Link aggregation operates as a sublayer between the MAC client and the MAC layer.

Each MAC passes received frames up for control or collection in an aggregator—a logical MAC that aggregates several links together. The MAC client sends frames to the aggregator for distribution among MACs, as follows:

Figure 11. Link Aggregation Traffic Flow



The aggregator and each MAC share the same MAC address, which means the MAC has no need to parse two different unicast MAC addresses.

Frame distribution uses an algorithm to distribute frames among MACs that prevents both the mis-ordering of frames belonging to the same "conversation," and frame duplication.

Link Aggregation Control

One port in an aggregation group is configured as a master so that all traffic (except control traffic) in the aggregation group logically passes through this port. It is recommended (although not required) that you set up the master first by CSP (task managing card/slot/ports), and unset last.

The following command creates link aggregation group *N* with port *slot#/port#* as master. Only one master port is allowed for a group. N must be in the range of [1...1023].

```
configure
  port ethernet <slot#/port#>
     link-aggregation master group <N>
     exit
```

Important: Link Aggregation Control Protocol (LACP) starts running only when the Master port is enabled.

Use the following command to add a port as member of link aggregation group number *N* only if the master port is assigned. Otherwise, it is added to the group when the master port is assigned:

```
port ethernet <slot#/port#>
   link-aggregation member group <N>
   exit
```

Important: The VPN can only bind the master port, and a VLAN can only be created on the master port. VPN CLI and vpnmgr return a failure message if you attempt to bind to a link aggregation member port.

Two redundant line cards and their controlling PAC/PSC function as a system; this allows loopback addressing between vertical slots. Each system that participates in link aggregation has a unique system ID that consists of two bytes priority (where the lowest number (0) has the highest priority) and six bytes of MAC derived from the first port's MAC address. The following command sets the system priority used to form the system ID. P is a hex in the range [0x0000..0xFFFF]. The default is 0x8000.

```
card <slot#>
  link-aggregation system-priority <P>
```

Ports in a system are assigned keys. The group number maps directly to the key, whereupon only ports with the same key can be aggregated. Ports on each side of the link use a different aggregation key.

The system ID, port key and port ID of two peers form the Link Aggregation Group Identifier (LAGID). You can aggregate links having the same LAGID. Systems are often configured initially with each port in its own aggregation (requiring a separate key per port), or with all ports in the same aggregation (a single key for all ports). Negotiation via LACP would qualify the actual aggregation.

Systems exchange information about system ID, port key and port ID with peers across the physical links using LACP.

LACP packets are defined with the Slow Protocol format. Each system sends out its own ("actor") information and its last received information about its peer ("partner") over the physical link.

Use the following commands to set the LACP parameters. LACP can run in active mode to send LACP packets periodically, or in passive mode, in which it only responds to LACP packets it receives.

LACP can send packets at either a slow (30s) or fast (1s) rate. The defaults for this release are **Active** and **Slow**; see the sample configuration below:

```
config
  port ethernet <slot#/port#>
     link-aggregation lacp active rate fast
```

Peers send out LACP packets when the state changes or if a difference is found from a received LACP packet about its own state.

Corresponding ports on a QGLC line card redundant pair cannot be active at the same time. Redundant ports share the same MAC address, so after a failover is resolved, the original port rejoins the link aggregation group.

Chapter 5
Configuring Management Settings

This chapter provides instructions for configuring management options such as Object Request Broker Element Management (ORBEM) and Simple Network Management Protocol (SNMP).

Configuring ORBEM Settings Overview

The system can be managed by a Common Object Broker Request Architecture (CORBA)-based software application called the Web Element Manager.

In order for the system to communicate with the server running the Web Element Manager application, you mus configure the ORBEM settings.

Important: Commands used in the configuration samples in this section provide base functionality. The most common commands and keyword options are presented. In many cases, other optional commands and keyword options are available. Refer to the *Command Line Interface Reference* for information about all commands.

Configure the system to communicate with the Web Element Manager:

Step 1 Set client ID parameters and configure the STOP/TCP port settings by applying the example configuration in the Configuring Client and Port Parameters section.

Step 2 Configure Internet Inter-ORB Protocol (IIOP) transport parameters by applying the example configuration in the Configuring Internet Inter-ORB Protocol (IIOP) Transport Parameters section.

Step 3 View your new ORBEM configuratioini by following the steps in the Verifying ORBEM Parameters section.

Step 4 Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Client and Port Parameters

Use the following example to set client ID parameters and configure the SIOP/TCP port settings:

```
configure
   orbem
      client id encrypted password <password>
      max-attempt <number>
     session-timeout <time>
      siop-port <port_number>
      event-notif-siop-port <siop_notif_port>
      event-notif-service
      end
```

Notes:

- You can issue the client id command multiple times to configure multiple clients.
- If a client ID is de-activated due to reaching the configured maximum number of attempts, use the **activate client id** command to reactivate it.
- If a firewall exists between the system and the Web Element Manager, open the SIOP port number and the TCP port number 15011.
- If the ORB Notification Service is enabled (event-notif-service), you can configure filters to determine which events are to be sent. By default, the Service sends all error and higher level events, “info” level events for the ORBS facility, CLI command logs, and license change logs. Optionally, configure a filter by including the **event-notif-service filter** command. Enter this command for each filter you need to configure.

Configuring Internet Inter-ORB Protocol (IIOP) Transport Parameters

Use the following example to configure IIOP transport parameters that enable ORB-based management to be performed over the network:

```
configure
   orbem
      iiop-transport
      iiop-port <iiop_port_number>
      event-notif-iiop-port <iiop_notif_port>
      end
```

Notes:

- If you are using the Secure Sockets Layer (SSL) option, do not enable the IIOP transport parameter. The Web Element Manager's default process enforces SSL.

Verifying ORBEM Parameters

Step 1 Enter the following to verify that the client was configured properly:

```
show orbem client table
```

This command lists the configured ORBEM clients and displays their state and privileges.

Step 2 Verify the ORBEM parameter configuration by entering the following command:

```
show orbem status
```

The following displays a sample of this command's output.

```
Service State : On
Management Functions : FCAPS
IOP URL : 192.168.1.150
SSL Port : 14131
TCP Port : 14132
Notification SSL Port : 7777
Notification TCP Port : 7778
Session Timeout : 86400 secs
Max Login Attempts : 5
IIOP Transport : On
Notification : On
Debug Level : Off
IDL Version Check : On
Number of Current Sessions : 1
Number of Event Channels Open : 0
Number of Operations Completed : 2895
Number of Events Processed : 0
Avg Operation Processing time : 87214 usecs
```

```
(last 1000) : 87950 usecs
```

Configuring System SNMP Settings

The system uses the SNMP to send traps or events to the Web Element Manager server or an alarm server on the network.

In order for the system to communicate with those devices, configure SNMP settings.

Important: Commands used in the configuration samples in this section provide base functionality. The most common commands and keyword options are presented. In many cases, other optional commands and keyword options are available. Refer to the *Command Line Interface Reference* for complete information.

To configure the system to communicate with the WEM server or an alarm server, enter:

```
[local]host_name#
```

Step 1 Configure SNMP parameters such as UDP port, and alarm server target by applying the example configuration in the Configuring SNMP and Alarm Server Parameters section.

Step 2 To view your new SNMP configuration, follow the steps in the Verifying SNMP Parameters section.

Step 3 Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring SNMP and Alarm Server Parameters

Use the following example to set SNMP and alarm server parameters:

```
configure
   system contact <contact_name>
   system location <location_name>
  snmp authentication-failure-trap
   snmp community <community_string>
   snmp server port <port_number>
  snmp target <name ip_address>
  snmp engine-id local <id_string>
   snmp notif-threshold <value> low <low_value> period <time_period>
  snmp user <user_name>
   end
```

Notes:

- The system contact is the name of the person to contact when traps are generated that indicate an error condition.
- A community string is a password that allows access to system management information bases (MIBs).
- The system can send SNMPv1, SNMPv2c, or SNMPv3 traps to numerous target devices. However, the Web Element Manager can only process SNMP version 1 (SNMPv1) and SNMP version 2c (SNMPv2c) traps. If the SNMP target you are configuring is the Web Element Manager application, configure this command to use version 1 or version 2c. Issue this command as many times as you need to configure multiple targets. If you configure multiple targets, generated alarms are sent to every configured target.
- The **`snmp engine-id local`** command is optional. It is only required if your network requires SNMP v3 support. The engine ID uniquely identifies the SNMP engine and the SNMP entitiy(ies) thus providing a security association between the two for the sending and receiving of data.
- SNMP user name is for SNMPv3 and is optional. There are numerous keyword options associated with this command.

Important: SNMPv3 traps are not supported for Web Element Management application.

Verifying SNMP Parameters

Step 1 To verify that the SNMP server information is correctly configured, enter the following command:

```
show snmp server
```

The following displays a sample output.

```
SNMP Server Configuration:
 Server State : enabled
 SNMP Port : 161
 sysLocation : chicago
 sysContact : admin
 authenticationFail traps : Enabled
 EngineID : 123456789
 Alert Threshold : 100 alerts in 300 seconds
 Alert Low Threshold : 20 alerts in 300 seconds
```

Step 2 Verify that the SNMP community(ies) were configured properly by entering the following command:

```
show snmp communities
```

The output of this command lists the configured SNMP communities and their corresponding access levels.

Step 3 Verify that the SNMP transports are configured properly by entering the following command:

```
show snmp transports
```

The following displays a sample output:

```
Target Name: rms1
IP Address: 192.168.1.200
Port: 162
Default: Default
Security Name: public
Version: 1
Security:
View:
Notif Type: traps
```

Controlling SNMP Trap Generation

The system uses SNMP traps to indicate that certain events have occurred. Refer to the *SNMP MIB Reference* for a complete listing of the traps supported by the system and their descriptions.

By default, the system enables the generation of all traps. However, you can disable individual traps to allow only traps of a certain type or alarm level to be generated. This section provides instructions for disabling/enabling SNMP traps.

Important: Commands used in the configuration samples in this section provide base functionality. The most common commands and keyword options are presented. In many cases, other optional commands and keyword options are available. Refer to the *Command Line Interface Reference* for complete information regarding all commands.

To configure SNMP trap generation:

Step 1 Set parameters by applying the following example configuration:

```
Configure
   snmp trap suppress
   snmp trap suppress <trap_name1> <trap_name2> ... <trap_nameN>
```

Important: If at a later time you wish to re-enable a trap that was previously suppressed, use the snmp trap enable command. Use the following command to specify which traps go to a specific trap server:

```
snmp trap enable <trap_name1> <trap_name2> ... <trap_nameN> target
<target-name>
```

Step 2 Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Chapter 6
Verifying and Saving Your Configuration

This chapter describes how to save your system configuration.

Verifying the Configuration

You can use a number of commands to verify the configuration of your feature, service, or system. Many are hierarchical in their implementation and some are specific to portions of, or specific lines in, the configuration file.

Feature Configuration

In many configurations, you have to set and verify specific features. Examples include APN and IP address pool configuration. Using these examples, enter the following commands to verify proper feature configuration:

show apn all

The output displays the complete configuration for the APN. In this example, an APN called apn1 is configured.

```
access point name (APN): apn1

authentication context: test

pdp type: ipv4

Selection Mode: subscribed

ip source violation: Checked drop limit: 10

accounting mode: gtpp No early PDUs: Disabled

max-primary-pdp-contexts: 1000000 total-pdp-contexts: 1000000

primary contexts: not available total contexts: not available

local ip: 0.0.0.0

primary dns: 0.0.0.0 secondary dns: 0.0.0.0

ppp keep alive period : 0 ppp mtu : 1500

absolute timeout : 0 idle timeout : 0

long duration timeout: 0 long duration action: Detection

ip header compression: vj

data compression: stac mppc deflate compression mode: normal

min compression size: 128

ip output access-group: ip input access-group:

ppp authentication:

allow noauthentication: Enabled imsi
```

```
 authentication:Disabled
```

Enter the following command to display the IP address pool configuration:

show ip pool

The output from this command should look similar to the sample shown below. In this example, all IP pools were configured in the *isp1* context.

```
context : isp1:

+-----Type: (P) - Public (R) - Private

 | (S) - Static (E) - Resource

 |

 |+----State: (G) - Good (D) - Pending Delete (R)-Resizing

 ||

 ||++--Priority: 0..10 (Highest (0) .. Lowest (10))

 ||||

 ||||+-Busyout: (B) - Busyout configured

 |||| |||||| vvvvv Pool Name Start Address Mask/End Address Used Avail

 ----- --------- --------------- --------------- -------- --------

PG00 ipsec 12.12.12.0 255.255.255.0 0 254 PG00

pool1 10.10.0.0 255.255.0.0 0 65534 SG00

vpnpool 192.168.1.250 192.168.1.254 0 5 Total Pool Count: 5
```

Important: To configure features on the system, use the *show* commands specifically for these features. Refer to the *Cisco Systems ASR 5000 Command Line Interface Reference* for more information.

Service Configuration

Verify that your service was created and configured properly by entering the following command:

show *<service_type> <service_name>*

The output is a concise listing of the service parameter settings similar to the sample displayed below. In this example, a P-GW service called pgw is configured.

```
Service name : pgw1

Service-Id : 1
```

```
Context : test1

Status : STARTED

Restart Counter : 8

EGTP Service : egtp1

LMA Service : Not defined

Session-Delete-Delay Timer : Enabled

Session-Delete-Delay timeout : 10000(msecs)

PLMN ID List : MCC: 100, MNC: 99

Newcall Policy : None
```

Context Configuration

Verify that your context was created and configured properly by entering the following command:

show context name *<name>*

The output shows the active context. Its ID is similar to the sample displayed below. In this example, a context named *test1* is configured.

```
Context Name     ContextID     State

------------     ---------     -----

test1            2          Active
```

System Configuration

Verify that your entire configuration file was created and configured properly by entering the following command:

show configuration

This command displays the entire configuration including the context and service configurations defined above.

Finding Configuration Errors

Identify errors in your configuration file by entering the following command:

show configuration errors

This command displays errors it finds within the configuration. For example, if you have created a service named "service1", but entered it as "srv1" in another part of the configuration, the system displays this error.

You must refine this command to specify particular sections of the configuration. Add the **section** keyword and choose a section from the help menu:

show configuration errors section ggsn-service

or

show configuration errors section aaa-config

If the configuration contains no errors, an output similar to the following is displayed:

```
##############################################################################

Displaying Global
AAA-configuration errors
##############################################################################

Total 0 error(s) in this section !
```

Saving the Configuration

Save system configuration information to a file locally or to a remote node on the network. You can use this configuration file on any other systems that require the same configuration.

Files saved locally can be stored in the SMC's CompactFlash or on an installed PCMCIA memory card on the SMC. Files that are saved to a remote network node can be transmitted through FTP, or TFTP.

Saving the Configuration on the Chassis

These instructions assume that you are at the root prompt for the Exec mode:

```
[local]host_name#
```

To save your current configuration, enter the following command:

```
save configuration url [-redundant] [-noconfirm] [showsecrets] [verbose]
```

Keyword/Variable	Description
url	Specifies the path and name to which the configuration file is to be stored. *url* may refer to a local or a remote file. *url* must be entered using one of the following formats: • **{ /flash | /pcmcia1 | /pcmcia2 }** *[/dir] /file_name* • **file:/{ /flash | /pcmcia1 | /pcmcia2 }** *[/dir] /file_name* • **tftp://**{ *ipaddress*|*host_name*[*:port#*]}[*/directory*]*/file_name* • **ftp://**[*username*[*:pwd*] *@*]{ *ipaddress*|*host_name*}[*:port#*][*/directory*] */file_name* • **sftp://**[*username*[*:pwd*] *@*]{ *ipaddress*|*host_name*}[*:port#*][*/directory*] */file_name* **/flash** corresponds to the CompactFlash on the SMC. **/pcmcia1** corresponds to PCMCIA slot 1. **/pcmcia2** corresponds to PCMCIA slot 2. *ipaddress* is the IP address of the network server. *host_name* is the network server's *hostname*. *port#* is the network server's logical port number. Defaults are: • tftp: 69 - data • ftp: 20 - data, 21 - control • sftp: 115 - data Note: *host_name* can only be used if the **networkconfig** parameter is configured for DHCP and the DHCP server returns a valid nameserv er.dx *username* is the username required to gain access to the server if necessary. *password* is the password for the specified username if required. */directory* specifies the directory where the file is located if one exists. */file_name* specifies the name of the configuration file to be saved. Note: Configuration files should be named with a .cfg extension.
-redundant	Optional: This keyword directs the system to save the CLI configuration file to the local device, defined by the url variable, and then automatically copy that same file to the like device on the Standby SMC, if available. Note: This keyword will only work for like local devices that are located on both the active and standby SMCs. For example, if you save the file to the /pcmcia1 device on the active SMC, that same type of device (a PC-Card in Slot 1 of the standby SMC) must be available. Otherwise, a failure message is displayed. Note: If saving the file to an external network (non-local) device, the system disregards this keyword.
-noconfirm	Optional: Indicates that no confirmation is to be given prior to saving the configuration information to the specified filename (if one was specified) or to the currently active configuration file (if none was specified).

Keyword/Variable	Description
showsecrets	Optional: This keyword saves the CLI configuration file with all passwords in plain text, rather than their default encrypted format.
verbose	Optional: Specifies to display every parameter that is being saved to the new configuration file.

Important: The **`-redundant`** keyword is only applicable when saving a configuration file to local devices. This command does not synchronize the local file system. If you have added, modified, or deleted other files or directories to or from a local device for the active SMC, you must synchronize the local file system on both SMCs.

To save a configuration file called system.cfg to a directory that was previously created called cfgfiles on the SMC's CompactFlash, enter the following command:

```
save configuration /flash/cfgfiles/system.cfg
```

To save a configuration file called simple_ip.cfg to a directory called host_name_configs, using an FTP server with an IP address of 192.168.34.156, on which you have an account with a username of administrator and a password of secure, use the following command:

```
save configuration
ftp://administrator:secure@192.168.34.156/host_name_configs/
simple_ip.cfg
```

To save a configuration file called init_config.cfg to the root directory of a TFTP server with a hostname of config_server, enter the following command:

`save configuration` *`tftp://config_server/init_config.cfg`*

Chapter 7
System Element Configuration Procedures

This chapter presents interface and port configurations procedures. Before beginning these procedures, refer to your product-specific administration guide for configuration information for your product.

This chapter includes the following:

Important: Make sure at least onePacket Accelerator Card (PAC) or Packet Services Card (PSC) is active before you configure system elements. Refer to *Configuring System Settings* in this guide for information and instructions on activating PACs/PSCs.

Creating Contexts

Even though multiple contexts can be configured to perform specific functions, they are all created using the same procedure.

Important: Commands used in the configuration examples in this section provide base functionality to the extent that the most common or likely commands and/or keyword options are presented. In many cases, other optional commands and/or keyword options are available. Refer to the *Command Line Interface Reference* for complete information regarding all commands.

To create a context, apply the following example configuration:

```
configure
   context <name>
      end
```

Notes:

- Repeat to configure additional contexts.

  Important: We recommend that if your system is using Ethernet 10/100 Line Cards, at least one context be configured per physical port in order to ensure adequate bandwidth for subscriber sessions.

Viewing and Verifying Contexts

Step 1 Verify that your contexts were successfully created by entering the following command:

```
show context all
```

The output is a two-column table that looks similar to the example below. In this example, it shows that two contexts were created: one called source and one called destination.

```
Context Name            ContextID                    State
------------            ---------                    -----
local                       1                        Active
source                      2                        Active
destination                 3                        Active
```

The left column lists the contexts that are currently configured. The center column lists the corresponding context ID for each of the configured contexts. The third column lists the current state of the context.

Step 2 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Step 3 Now that the context has been created, interfaces and specific functionality can be configured within the context, proceed to any of the other sections in this chapter for instructions on configuring specific services and options.

Creating and Configuring Ethernet Interfaces and Ports

Regardless of the type of application interface, the procedure to create and configure it consists of the following:

Step 1 Create an interface and assign an IP address and subnet mask to it by applying the example configuration in the Creating an Interface

Step 2 Assign a physical port for use by the interface and bind the port to the interface by applying the example configuration in the Configuring a Port and Binding it to an Interface section.

Step 3 Optionally configure a static route for the interface by applying the example configuration in the Configuring a Static Route for an Interface section.

Step 4 Repeat the above steps for every individual interface to be configured.

Important: This section provides the minimum instruction set for configuring interfaces and ports to allow the system to communicate on the network. Commands that configure additional interface or port properties are provided in the *Ethernet Port Configuration Mode* and *Ethernet Interface Configuration Mode* chapters of the *Command Line Interface Reference.*

Caution: To ensure that system line card and port-level redundancy mechanisms function properly, the Spanning Tree protocol must be disabled on devices connected directly to any system port. Failure to turn off the Spanning Tree protocol may result in failures in the redundancy mechanisms or service outage.

Creating an Interface

Use the following example to create a new interface in a context:

```
configure
   context <name>
      interface <name>
         ip address <address subnetmask> [ secondary ]
         end
```

Notes:

- *Option:* Add the **loopback** keyword option to the **interface** *<name>* command, to set the interface type as “loopback”.
- *Option:* Add the **secondary** keyword to the **ip address** command, to assign multiple IP addresses to the interface. IP addresses can be in IPv4 or IPv6 format.

- *Option:* In the interface config mode, add the **port-switch-on-L3-fail address** command, to configure the interface for switchover to the port on the redundant line card if connectivity to a specified ip address is lost. This can be in IPv4 or IPv6 format.

Configuring a Port and Binding It to an Interface

Use the following example configuration to configure and assign a port to an interface:

```
configure
    port ethernet <slot#/port#>
      description <description>
      no shutdown
      bind interface <interface_name>  <context_name>
      end
```

Notes:

- For **port ethernet** *slot#*, use the actual chassis slot in which the line card is installed. This could be any number from 17 to 23, or 26 to 39, or 42 to 48.
- *Option:* In the Ethernet Port configuration mode, add the preferred slot slot# command if LC port redundancy was enabled at the card level and you want to specific a port preference.
- *Option:* In the Ethernet Port configuration mode, configure the port speed, if needed, by entering the **medium** command. Refer to the *Command Line Interface Reference* for a complete explanation of this command.
- Binding associates the port and all of its settings to the interface.

Configuring a Static Route for an Interface

Use the following example to configure a static route for an interface:

```
configure
    context <name>
       ip route <ip_address> <netmask> next-hop <gw_address> <interface_name>
       end
```

Notes:

- *ip_address* and *netmask* are the IP address and subnet mask of the target network.
- *gw_address* is the IP address of the default gateway or next-hop route.
- To configure a route to the gateway router, use 0.0.0.0 for the network and mask variables.

- Repeat as needed. Multiple static routes can be configured to the same destination to provide an alternative means of communication in case the preferred route fails.

Viewing and Verifying Port Configuration

Step 1 Verify that your interface configuration settings are correct by entering the following commands:

context *<context_name>*

show ip interface

context_name represents the name of the context in which the interface was created. The output from these commands should look similar to that displayed in the following example.

In this example an interface named mgmt1 was configured in the local context.

```
Intf Name:       mgmt1
Intf Type:       Broadcast
IP State:        UP (Bound to 17/1 untagged, ifIndex 285278209)
IP Address:      192.168.100.3        Subnet Mask: 255.255.255.0
Bcast Address:   192.168.100.255      MTU:             1500
Resoln Type:     ARP                  ARP timeout:     3600 secs
L3 monitor LC-port switchover:  Disabled
Number of Secondary Addresses:  0
Total interface count:          1
```

Step 2 Verify that your port configuration settings are correct by entering the following command:

show configuration port *<slot#/port#>*

slot# is the chassis slot number of the line card on which the physical port resides. *slot#* can be any integer value from 17 to 39, and 42 to 48.

This command produces an output similar to that displayed in the following example that shows the configuration for port 1 on the line card installed in chassis slot 17.

In this example, the port is bound to an interface called rp1 configured in a context called source.

```
config
   port ethernet 17/1
```

```
description LC17/1_RP1
no shutdown
bind interface rp1 source
#exit   end
```

Step 3 Verify that your static route(s) was configured properly by entering the following command:

`show ip static-route`

This command produces an output similar to that displayed in the following example that shows a static route to a gateway with an IP address of 192.168.250.1

```
Destination      Nexthop          Protocol  Prec  Cost  Interface
0.0.0.0/0        192.168.250.1    Static      0     0     SPIO1
0.0.0.0/0        192.168.250.1    Static      0     0     rp1
source
```

Step 4 Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Creating and Configuring ATM Interfaces and Ports

This section describes the minimum configuration required to use IP over ATM through an Optical ATM line card (OLC). The procedures define how to create and to configure the IP interfaces, ATM interfaces, and the ATM ports:

Step 1 Set the framing method for a specific OLC-type line card and make the card "active" by using the procedure defined in Enabling the OLC (ATM) Line Card .

Step 2 Create an IP over ATM interface (PVC interface) by following the example configuration in the Creating an IP Interface for Use with an ATM Port section.

Step 3 Enable the ATM port, create the IPoA (PVC) interface for the ATM port, and enable the PVC by applying the example configuration in the Configuring an ATM Port to Use an IP Interface section.

Steps 2 and 3 together configure the data plane.

Step 4 Configure an ATM port to use with an SS7 link ID by applying the example configuration in the Configuring an ATM Port for an SS7 Link section.

Step 4 configures the control plane through an SS7/IPoA (PVC) interface.

> ***Important:*** Do not attempt to bind the link at this time. Complete the rest of the procedure (steps 5, 6, and 7) and return to bind the link to the port. The SS7 link can only be bound to the ATM port after the configuration for the SS7 routing domain has been completed as described in the *3G SGSN Configuration* section of the *SGSN Administration Guide*.

Step 5 Configure the appropriate timing source (BITS from the SPIO or line-timing from attached remote) to ensure transmit synchronization by applying the example configuration in *Configuring Transmit Timing Source* section in the *Configuring System Settings* chapter.

Step 6 Verify the port and interface configuration with the procedure Verifying Port and Interface Configuration .

Step 7 Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Enabling the OLC (ATM) Line Card

Use the following example to select an OLC line card and set the framing type:

```
configure
   card <slot#>
      framing <SDH|SONET>
      no shutdown
      end
```

Notes:

- The default framing type is SONET for an Optical (ATM) line card.
- Setting the framing method is required to make the card operational.
- Entering **no shutdown** makes the card active.

Creating an IP Interface for Use with an ATM Port

Use the following example to create an IP interface to use with ATM:

```
configure
   context <ctxt_name>
      interface <intf_name> point-to-point
         ip address <ip_addr> <net_mask>
         ip address <ip_addr> <net_mask> secondary
         end
```

Notes:

- The context must be the one in which you have or will configure the SGSN service.
- You must enter the **point-to-point** keyword to create the PVC (Permanent Virtual Connection) interface for the IP over ATM.

Configuring an ATM Port to Use an IP Interface

Use the following example to configure an ATM port to use with an IP interface:

```
configure
   port atm <slot#>/<port#>
      no shutdown
      pvc vpi <vpi_num> vci <vci_num>
         no shutdown
         bind interface <ifc_name> <ctx_name>
         end
```

Notes:

- The context must be the one you used when creating the IP interface (PVC) for the ATM port.

Configuring an ATM Port for an SS7 Link

Use the following example to configure an ATM port to use with an SS7 link:

```
configure
   port atm <slot#>/<port#>
      no shutdown
      pvc vpi <vpi_num> vci <vci_num>
         no shutdown
         end
```

Notes:

- The PVC for the SS7 link has been created but can not be bound unless the SS7 routing domain configuration has already been completed (see the *SGSN Administration Guide*).
- Complete optional ATM port configuration (see the *ATM Port Configuration Mode* in the *Command Line Interface Reference*) and the other steps in this procedure to set timing and save the configuration.

Binding an SS7 Link to an ATM Port

Use the following example to bind an already configured SS7 link to a PVC interface for an ATM port:

```
configure
   port atm <slot#>/<port#>
      pvc vpi <vpi_num> vci <vci_num>
         bind link ss7-routing-domain <ss7rd_id> linkset-id <id> link-id <id>
         end
```

Notes:

- Save the configuration as described in the *Saving Your Configuration* chapter.

Verifying Port and Interface Configuration

Step 1 Verify that your interface configuration settings are correct by entering the following commands:

```
context <context_name>
show ip interface
```

context_name represents the name of the context in which the interface was created. The output from these commands should look similar to that displayed in the following example.

In this example an interface named mgmt1 was configured in the local context.

```
Intf Name:       ipoa

Intf Type:       Point to point

IP State:        UP (Bound to 31/1 untagged, ifIndex 285278209)

IP Address:      192.168.100.3        Subnet
Mask:     255.255.255.0

Bcast Address:   192.168.100.255      MTU:             1500

Resoln Type:     ARP                  ARP timeout:     3600 secs

Number of Secondary Addresses:  0

Total interface count:  1
```

Step 2 Verify that your port configuration settings are correct by entering the following command:

show configuration port *<slot#>/<port#>*

This command produces an output *similar* to that displayed in the following example:

```
config

   port atm 31/1

      no shutdown

      pvc vpi 121 vci 4444

          no shutdown

          bind interface ipoa sgsn3g

       #exit

    #exit

 end
```

Creating and Configuring Frame Relay Interfaces and Ports

This section shows the minimum configuration required to configure a frame relay interface on a channelized line card. To create and configure the frame relay interfaces and ports:

Step 1 Select a channelized line card (CLC) and set the framing method by applying the example configuration in the Setting the Characteristics of the Channelized Line Card section.

Step 2 Configure the path, framing, mapping, Frame Relay characteristics, and the data link connection identifier s(DLCIs) as illustrated in the example configuration in the Configuring the Channel Characteristics section.

Step 3 Configure the appropriate timing source (BITS from the SPIO or line-timing from attached remote) to ensure transmit synchronization by applying the example configuration in the Configuring Transmit Timing Source section in the *Configuring System Settings* chapter.

> ***Important:*** Before you can move to *Step 4* to bind a DLCI to a port, you must complete the link configuration by configuring Peer-NSEIs and/or SS7 routing domains as described in the *SGSN Service Configuration Procedures* chapter of the *SGSN Administration Guide*. Return to this procedure when your link configuration is complete.

Step 4 Bind the link to the port by applying the example configuration in the section for Binding a DLCI .

Step 5 Verify the card, port and link configuration and status with the commands illustrated in

Step 6 Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Setting the Characteristics of the Channelized Line Card

Use the following example to set the operational characteristics, the framing type, the header type, the service type, and the boot time E1 framing type, for the Frame Relay Channelized Line Card (CLC):

```
configure
   card <slot>
      framing <framing_type> [ ds1 | e1 ]
      header-type { 2-byte | 4-byte }
      initial-e1-framing { crc4 | standard }
      service-type frame-relay
      no shutdown
      end
```

Notes:

- Make a note of the information you configure - you will need it again later for the **path** command used to configure channel characteristics.
- The default *framing_type* is SONET for the channelized line card.
- With releases 8.1 and higher, we recommend that you include the signal type, either **ds1** or **e1**, when defining the framing.
- With releases 8.1 and higher, you need to set the **service-type** for the CLC card to *frame-relay*, all other options are not fully supported at this time.

Configuring the Channel Characteristics

Use the following example to configure the path, framing, mapping, timeslots, and the Frame Relay interface and LMI characteristics for a specific CLC port:

```
configure

   port channelized <slot#>/<port#>

      path <path_id> { ds1 | e1 } <number_of_connections> <frame_mapping>
<multiplex#> <multiplex#> framing <framing_mode> mapping-mode <mapping_mode> [
timeslots <timeslot#> [ <timeslot#> ] ]
```

Important: You should make a record of the path definition values you enter as the information will be needed again with other configuration commands.

```
      frame-relay path <path_id> { ds1 | e1 } <number_of_connections> [ timeslot
<timeslot#> [ intf-type  <intf_type> [ lmi-type <lmi_type> ] ] ]

      dlci path <path_id> { ds1 | e1 } <number_of_connections> { dlci <dlci_id>
| timeslot <timeslot#> }

      no shutdown

      end
```

Binding a DLCI

Use the following procedure to bind the DLCI to the channelized (Frame Relay) port.

Important: This procedure should not be attempted until after the configuration is completed for Peer-NSEIs and/or SS7 routing domains as described in the *SGSN Administration Guide*.

```
configure

   port channelized <port#>
```

```
        bind link { peer-nsei <nse_id> ns-vc-id <nsvc_id> | ss7-routing-domain
<ss7rd_id> linkset-id <id> link-id <id>}

        end
```

Verifying the Frame Relay Interface Configuration and Status

Several commands generate display outputs that provide information about the Frame Relay card, port, DLCI and link configurations. The most commonly used commands are illustrated below. All of these commands are issued from the Exec mode.

Display Port and DLCI Configuration Details

```
[local]<hostname>#show port info 27/1

Port: 27/1

  Port Type                    : STM1/OC3 Channelized

  Description                  : (None Set)

  Controlled By Card           : 11 (Packet Services Card)

  Redundancy Mode              : Port Mode

  Framing Mode                 : SDH

  Redundant With               : Not Redundant

  Preferred Port               : Non-Revertive

  Physical ifIndex             : 453050368

  Administrative State         : Disabled

  Link State                   : Unknown

  Line Timing                  : Yes

  SFP Module                   : Not Present

  Path 1 e1 1                  : tu12-au3 1/1 crc4 bit-async

    Timeslots                  : 12-14

    Frame Relay Intf Typ       : DCE

    Frame Relay LMI Type       : Q933A

    Frame Relay LMI n391       : 6

    Frame Relay LMI n392       : 2
```

```
    Frame Relay LMI n393       : 2
    Frame Relay LMI t391       : 10
    Frame Relay LMI t392       : 15
      Frame Relay DLCI 243
        Logical ifIndex        : 453050369
        Admin State            : Disabled
        Operational State      : Down, Standby
      Shaping                  : WFQ: Weight 1
      Number of DLCI           : 1
      Reserved Bandwidth       : 0 of 192000 bits/sec
  Path 1 e1 2                  : Unused
  Number of DLCI               : 1
```

Display Port and DLCI Configuration and Status

The following display is only a partial output to illustrate the channelized port and DLCIs:

show port table

```
Port  Type                    Admin     Oper  Link  State     Redundant With
----  ----------------------  --------  ----  ----  --------  --------------
27/1  STM1/OC3 Channelized    Enabled    -    Up      -        None
      FR DLCI   1  1  1  52   Enabled   Up    -     Active     -
      FR DLCI   1  2  1  53   Enabled   Down  -     Active     -
```

Chapter 8
Software Management Operations

This chapter provides information about software management operations on the system. Software management sections in this chapter include:

Understanding the Local File System

The Switch Processor Card (SPC)/System Management Card (SMC) provides control and management for the system.

The local file system is made up of files that are stored on one or more of the following:

- **`/flash`** - A CompactFlash card, located on the circuit board of the SPC/SMC, is the default storage media for the operating system software image, CLI configuration, and crash log files used by the system.
- **`/pcmcia1`** - This device is available when an ATA Type I or Type II PCMCIA card is inserted into PC-Card Slot 1 (upper slot) on the front panel of the SPC/SMC.
- **`/pcmcia2`** - This device is available when an ATA Type I or Type II PCMCIA card is inserted into PC-Card Slot 2 (lower slot) on the SPC's front panel. Note that this option is not available for use with the SMC.
- /hd-raid: This device is the hard drive installed on the SMC. Disk names "hd-local1" and "hd-remote1" are used on ASR 5000s. An XFS-formatted RAID disk is mounted on "/mnt/hd-raid". Users can gain access to part of it from either "/hd-raid" or "/mnt/user/hd-raid".

> ***Important:*** For this release, local filesystem access is to master SMC only.

File Types Used by the Local File System

The following file types can be located in the local file system:

- **Operating System Software Image File**: This binary file type is identified by its **.bin** extension. The file is the operating system that is loaded by the system upon startup or reloading. This is an executable, read-only file that cannot be modified by end users.

> ***Important:*** The ST16 and ASR 5000 platforms require different software image files. The name of the image file indicates whether it is intended for use on an ST16 or an ASR 5000. A file intended for use on an ST16 uses the convention xxxxx.st16.bin where xxxxx is the software build information. Alternatively, a file intended for use on an ASR 5000 uses the convention xxxxx.asr5000.bin.

- **CLI Configuration File**: This file type is identified by its**.cfg** extension. These are text files that contain CLI commands that work in conjunction with the operating system software image. These files determine services to be provided, hardware and software configurations, and other functions performed by the system. The files are typically created by the end user. You can modify the files both on and off-line and use descriptive long filenames.
- **System File**: Only one file identified by a **.sys** extension is used by the system. The boot.sys file contains system-specific information, which describes how the system locates, and in what priority it loads, file groups (paired .bin and .cfg files) from its boot stack.

- **Abridged Crash Log**: The abridged crash log, identified by its **crashlog** filename, contains summary information about software or hardware failures that occur on the system. This file is located in the **/flash/crsh2/** directory on the device. You can view the contents of this file through the CLI, but you cannot modify the file.

Understanding the boot.sys File

The system uses the boot.sys file to store the prioritized boot stack parameters and file groups the system uses during startup. Modify this file only through CLI commands and not through external means. Boot parameters contain information the system needs to locate the operating system image file, including:

- **bootmode**: This setting is typically configured to normal, and identifies how the system starts.
- **network interface configuration**: Use these optional boot method settings when you configure the system to obtain its operating system image from an external network server that is using one of the management LAN interfaces on the SPIO card.
- **terminal-speed configuration**: This parameter identifies the data transfer rate at which a serial interface communicates on the console port. The default setting for this parameter is 115200 bps (115.2 Kbps). You can change this and other settings with RS-232 Port Configuration Mode commands.
- **boot stack information**: The boot stack is made up of prioritized file group entries that designate the operating system image file and the CLI configuration file to load.

When a system is unpacked and started for the first time, the boot.sys file is configured to use the normal boot mode and load the operating system software image from the /flash directory.

There is no CLI configuration file contained on the local file system. This causes the system to automatically start its CLI-based Quick Setup Wizard upon the first successful boot. Refer to Getting Started for more information on using the Quick Setup Wizard.

Maintaining the Local File System

Use CLI commands to manage and maintain the devices that make up the local file system. Execute all the commands described in this section in the Exec Mode. Unless otherwise specified, you must have security administrator or administrator privileges to execute these commands.

File System Management Commands

Use the commands in this section to manage and organize the local file system.

Synchronizing the File System

Commands are supported for mirroring the local file systems from the active SPC/SMC to the standby SPC/SMC in systems containing two cards. Use these commands to synchronize any or all of the local devices.

Important: Crash log files are not synchronized when these commands are executed.

The following command synchronizes the file systems between two SPCs:

```
card spc synchronize filesystem {/flash|/pcmcia1|/pcmcia2|all}
[checkonly] [ reverse]} [-noconfirm]
```

The following command synchronizes the file systems between two SMCs:

```
card smc synchronize filesystem {/flash|all} [checkonly] [reverse]} [-
noconfirm ]
```

Table 8. Command Syntax Descriptions

Keyword/Variable	Description
`/flash`	Synchronizes only the CompactFlash file system on the standby SPC/SMC.
`/pcmcia1`	Synchronizes only the file system of the PCMCIA card installed in the PCMCIA 1 slot on the standby SPC/SMC.
`/pcmcia2`	Synchronizes only the file system of the PCMCIA card installed in the PCMCIA 2 slot on the standby SPC.
`all`	Specifies that filesystems on all available matching local devices (be synchronized.
`checkonly`	Displays a list of files that can be synchronized without executing any synchronization actions.
`reverse`	Performs the specified operation on the standby SPC/SMC.
`-noconfirm`	Use this keyword to disable the “Are you sure? [Yes | No]” confirmation prompt asked before you execute the command.

Example
The following command synchronizes the file systems on two SPC /flash devices.

```
card spc synchronize filesystem /flash
```

Creating Directories

Use the **mkdir** command to create a new directory on the specific local device. This directory can then be incorporated as part of the path name for any file located in the local file system.

mkdir /flash/<*dir_name*>

For ST16s:

```
mkdir {/flash|/pcmcia1|/pcmcia2}/<dir_name>
```

For ASR 5000s:

```
mkdir {/flash|/pcmcia1|/hd-raid}/<dir_name>
```

Example
Use the following command to create a directory named *configs*:

```
mkdir /flash/configs
```

Renaming Files and Directories

Use the **rename** command to change the name of a file from its original name to a different name. Remember to use the same file extension, if applicable, to ensure that the file type remains unchanged.

For ST16s:

```
rename {/flash|/pcmcia1|/pcmcia2}/<src_filename>
{/flash|/pcmcia1|/pcmcia2}/<dst_filename> [-noconfirm]
```

For ASR 5000s:

```
rename {/flash|/pcmcia1|/hd-raid}/<src_filename> {/flash|/pcmcia1|/hd-
raid}/<dst_filename> [-noconfirm]
```

Table 9. rename command options

Keyword/Variable	Description
src_filename	The name of the source file, with its extension, that you are renaming.
dst_filename	The name of the destination file, with its extension, to which the source file is being renamed. Be sure to use the same file extension to ensure that the file type remains unchanged.

Keyword/Variable	Description
-noconfirm	Disables the "Are you sure? [Yes | No]" confirmation prompt, asked before executing the command.

Example
Use the following command to rename a file named *pdsn_test.cfg to pdsn_prod.cfg* on the */flash* local device.

```
rename /flash/pdsn_test.cfg /flash/pdsn_prod.cfg -noconfirm
```

Important: Use the **rename** command only within the same local device. You cannot rename a file and place it onto another local device at the same time. To move a renamed file, you must use the **copy** command.

Copying Files and Directories

The **copy** command copies files from one device to another device or location.

```
copy from_url to_url [-noconfirm]
```

Table 10. copy command parameters

Keyword/Variable	Description
-noconfirm	Disables the *"Are you sure? [Yes | No]"* confirmation prompt asked before executing the command.

Example
The following copies a file *test.cfg* from an external server using FTP to /pcmcia1.

```
copy ftp://root:network@192.168.1.151/system/test.cfg /pcmcia1/test.cfg
```

Deleting Files

The **delete** command removes a designated file from its specified location on the local file system. This command can only be issued to a local device on the SPC/SMC. Note that this command does not allow for wildcard entries; each filename must be specified in its entirety.

Caution: Do not delete the boot.sys file. If deleted, the system will not reboot on command and will be rendered inoperable.

For ST16s:

```
delete {/flash|/pcmcia1|/pcmcia2}/<filename> [-noconfirm]
```

For ASR 5000s:

```
delete {/flash|/pcmcia1|/hd-raid}/<filename> [-noconfirm]
```

Table 11. delete command variables

Keyword/Variable	Description
filename	The name of the file, including any extension, that will be deleted.
-noconfirm	Disables the *"Are you sure? [Yes | No]"* confirmation prompt asked before executing the command.

Example
The following command deletes a file named *test.cfg* from the /pcmcia1 local device.

```
delete /pcmcia1/test.cfg
```

Deleting Directories

The **rmdir** command deletes a current directory on the specific local device. This directory can then be incorporated as part of the path name for any file located in the local file system.

Important: The directory you want to remove (delete) must be empty before executing the **rmdir** command. If the directory is not empty, the CLI displays a *Directory not empty* message and will not execute.

```
rmdir url/<dir_name>
```

Table 12. mkdir command options

Keyword/Variable	Description
dir_name	The name of the directory to be removed.
-noconfirm	Disables the *"Are you sure? [Yes | No]"* confirmation prompt asked before executing the command.

Example
The following command deletes an empty directory named *configs* on the /flash local device.

```
rmdir /flash/configs
```

Formatting Local Devices

The **format** command performs a low-level format of a local device. This operation formats the device to use the FAT16 formatting method, which is required for proper read/write functionality with the operating system.

Important: Local devices that have been formatted using other methods such as NTFS or FAT32 may be used to store various operating system, CLI configuration, and crash log files. However, if placing a new local device into the SPC/SMC for regular use, it is recommended that the device be formatted by the system prior to use. This ensures that the FAT16 file allocation table format is used, preventing any possible discrepancies between other formats used with other operating systems.

Caution: Use of the **format** command should be carefully monitored and approved by operations management personnel. Formatting a local device removes all files and information stored on the local device.

To format a local device for use by the local file system, enter the following command:

For ST16s:

```
format {/flash|/pcmcia1|/pcmcia2}
```

For ASR 5000s:

```
format {/flash|/pcmcia1|/hd-raid}
```

Applying Pre-existing CLI Configuration Files

A pre-existing CLI configuration file is any .cfg file created to provide utility functions (such as clearing all statistics during testing) or created off-line (such as using a text editor). There may be pre-existing configuration files stored on the local file system that can be applied to a running system at any time.

Caution: If a configuration file is applied to a system currently running another CLI configuration, any like contexts, services, logical interfaces, physical ports, IP address pools, or other configured items will be overwritten if the same command exists in the configuration file being applied. Take caution to ensure that you are knowledgeable of the contents of the file being applied and understand what the service ramifications are if a currently running command is overwritten. Also note that changes will not be saved automatically.

A CLI configuration file, or script containing CLI commands, can be applied to a running system by entering the following command at the Exec mode prompt:

```
configure url [ verbose ]
```

Table 13. `configure` command options

Keyword/Variable	Description
verbose	Displays each line and its line number when applying a pre-existing CLI configuration file or script.

Example
The following command applies a pre-existing CLI configuration file named clearcmds.cfg on the /flash local device.

```
configure /flash/clearcmds.cfg
```

Viewing Files on the Local File System

This section describes how to view a variety of files.

Viewing the Contents of a Local Device

The contents, usage information, and file system directory structure of any local device can be viewed by entering the following command at the Exec mode prompt:

For ST16s:

```
directory {/flash|/pcmcia1|/pcmcia2}
```

For ASR 5000:

```
directory {/flash|/pcmcia1|/hd-raid}
```

Viewing CLI Configuration and boot.sys Files

The contents of CLI configuration and boot.sys files, contained on the local file system, can be viewed off-line (without loading them into the OS) by entering the following command at the Exec mode prompt:

For ST16s:

```
show file url {/flash|/pcmcia1|/pcmcia2}/<filename>
```

For ASR 5000:

```
show file url {/flash|/pcmcia1|/hd-raid}/<filename>
```

Where *filename* is the name of the file, including any extension.

Important: Operator and inspector-level users can execute the **show file** command but can not execute the **directory** command.

Validating an Operating System File

The operating system software image file, identified by its .bin extension, is a non-readable, non-editable file that executes on the system, creating its runtime operating system (OS).

It is important to verify a new operating system image file before attempting to load it. To accomplish this, a proprietary checksum algorithm is used to create checksum values for each portion of the application stored within the .bin file during program compilation.

This information can be used to validate the actual file against the checksum values stored within the file during its compilation. If any portion of the image file has become corrupted (e.g. the file was truncated or was transferred using ASCII mode instead of binary mode, etc.), then this information is reported and the file is deemed unusable.

To validate an operating system software image file, enter the following command at the Exec mode prompt:

For ST16s:

show version {/flash|/pcmcia1|/pcmcia2} <*filename*> [**all**]

For ASR 5000:

show version {/flash|/pcmcia1|/hd-raid} <*filename*> [**all**]

The output of this command displays standard operating system version information, plus the exact file size and sub-component verification information for the entire .bin file.

If an invalid file is found, the system displays a failure message similar to these:

```
Failure: Image /flash/os_3888.bin CRC check failed!

Failure: /flash/OS.3819.bin, has a bad magic number
```

Configuring the Boot Stack

The boot stack consists of a prioritized listing of operating system software image-to-CLI configuration file associations. These associations determine the software image and configuration file that gets loaded during system startup or upon a reload/reboot. Though multiple associations can be configured, the system uses the association with the highest priority. In the event that there is an error processing this association (e.g. one of the files cannot be located), the system attempts to use the association with the next highest priority. Priorities range from 1 to 100, with 1 being the highest priority. The maximum number of boot stack entries that may be configured in the boot.sys file is 10.

Boot stack information is contained in the boot.sys file, explained earlier in the Understanding the boot.sys File section of this chapter. In addition to boot stack entries, the boot.sys file contains any configuration commands required to define the system boot method as explained in the section that follows.

System Boot Methods

The following methods are supported for loading and executing system software and configuration files on startup:

- **Local-Booting Method:** The default boot method that uses software image and configuration files stored locally on the system. Upon system startup or reboot, the system looks on one of its local devices (**`/flash`**, **`/pcmcia1`**, **`/pcmcia2`** (SPC only), or /hd-raid (SMC only)) located on the primary SPC/SMC for the specific software image and accompanying configuration text file.

When using the local-booting method, you only need to configure boot stack parameters.

- **Network-Booting Method:** The system can be configured to obtain its software image from a specific external network server while it is paired with a configuration text file that resides on the system. When using network booting, you need to configure the following:
  - Boot stack parameters, which define the files to use and in what priority to use them
  - Boot interface and network parameters defining the SPIO management LAN interface and the methods to use to reach the external network server
  - Network booting delay time and optional name server parameters defining the delay period (in seconds) to allow for network communications to be established, and the IP address of any Domain Name Service (DNS) name server that may be used

More detailed information on how to configure the system to use the network-booting method will be provided later in this chapter.

Viewing the Current Boot Stack

To view the boot stack entries contained in the boot.sys file enter the following:

```
show boot
```

Important: Operator and inspector-level users can execute the **show boot** command.

The example below shows the command output for a local booting configuration. Notice that in this example that both the image file (operating system software) and configuration file (CLI commands) are located on the **/flash** device.

```
boot system priority 18 image /flash/build15003.aaaa.bin \config
/flash/general_config.cfg

boot system priority 19 image /flash/build14489.bbbb.bin \config
/flash/general_config_3819.cfg

boot system priority 20 image /flash/build14456.cccc.bin \config
/flash/general_config_3665.cfg
```

The example below shows the output for a combination network booting and local booting configuration. Notice in this example that the first two boot stack entries (Priorities 18 and 19) load the image file (operating system software) from an external network server using the Trivial File Transfer Protocol (TFTP), while all configuration files are located on the **/flash** device.

Also notice the boot network interface and boot network configuration commands located at the top of the boot stack. These commands define what SPIO management LAN interface(s) to use and information about communicating with the external network server that hosts the operating system software image file.

```
boot interface spio-eth1 medium auto media rj45

boot networkconfig static ip address spio24 192.168.1.150 netmask
255.255.255.0

boot delay 15



boot system priority 18 image
tftp://192.168.1.161/tftpboot/build15003.st16.bin \config
/flash/general_config.cfg

boot system priority 19 image
tftp://192.168.1.161/tftpboot/build14489.st16.bin \config
/flash/general_config.cfg

boot system priority 20 image /flash/build14456.st16.bin \config
/flash/general_config.cfg
```

To identify the boot image priority that was loaded at the initial boot time enter:

show boot initial-config

The example below displays the output:

```
[local]host# show boot initial-config

Initial (boot time) configuration:

      image tftp://192.168.1.161/tftpboot/build15429.xxxx.bin \
```

```
config /flash/general_config.cfg
   priority 1
```

Adding a New Boot Stack Entry

Important: Before performing this procedure, verify that there are less than 10 entries in the boot.sys file and that a higher priority entry is available (i.e. that minimally there is no priority 1 entry in the boot stack). Refer to *Viewing the Current Boot Stack* for more information.

If priority 1 is in use, then you must renumber the existing entry(ies) to ensure that at least that priority is available. The maximum number of boot stack entries that can be contained in the *boot.sys* file is 10. If there are already 10 entries in the boot stack, you must delete at least one of these entries (typically, the lowest priority) and, if necessary, renumber some or all of the other entries before proceeding. Refer to *Deleting a Boot Stack Entry* for more information.

This procedure details how to add new boot stack entries to the *boot.sys* file. Make sure you are at the Exec mode prompt and enter the following commands:

```
configure
   boot system priority number image <image_url> config <cfg_url>
```

Table 14. boot system priority command options

Keyword/Variable	Description
number	Specifies the boot priority number for the file group (combination of operating system software image and CLI configuration file). This value must be entered as an integer, ranging from 1 to 100, with the lowest number having the highest boot priority.A existing priority number, used by another boot stack entry, may be entered. However, this will overwrite the existing entry in the boot.sys file.

Example
The following command creates a new boot stack entry, using a boot priority of *3*, an image file named *os_20000.XXX.bin*, and a configuration file named *general.cfg*.

```
boot system priority 3 image /flash/os_20000.XXX.bin config /flash/general.cfg
```

Important: Boot stack changes saved to the *boot.sys* file are not executed until the system is restarted.

Synchronize the local file systems on the SPCs/SMCs by entering one of the following commands:

For SPCs:

```
card spc synchronize filesystem all
```

For SMCs:

```
card smc synchronize filesystem all
```

Deleting a Boot Stack Entry

This procedure details how to remove an individual boot stack entry from the boot.sys file. Make sure you are at the Exec mode prompt and enter the following commands:

```
configure
   no boot system priority number
```

Where number specifies the boot priority used for the boot stack entry. This command removes that specific entry from the boot stack, causing the boot.sys file to be overwritten.

Network Booting Configuration Requirements

Configuring the Boot Interface

Boot interface parameters define the SPIO's management LAN interface that the system will use to communicate with the management network when using the network booting method.

This procedure details how to configure the boot interface for reliable communications with your network server. Make sure you are at the Exec mode prompt:

```
[local]host_name#
```

Step 1 Enter the Global Configuration mode by entering the following command:

```
configure
```

The following prompt appears:

```
[local]host_name(config)#
```

Step 2 Enter the following command:

```
boot interface {spio-eth1|spio-eth2} medium {auto|speed
{10|100|1000} duplex {full|half}} media {rj45|sfp}
```

Keyword/Variable	Description
interface	Specifies the desired SPIO interface to use when communicating with the network server during boot. **spio-eth1** corresponds to either the **RJ-45 1** or **SFP 1** interface on the SPIO. **spio-eth2** interface that corresponds to either the **RJ-45 2** or **SFP 2** interface on the SPIO. ***Important:*** Use SPIO port 1 for network booting.
medium	Specifies the speed that the interface should implement to communicate on the network. **auto** implements auto-negotiation to determine the highest possible speed and duplex mode. **speed** specifies the rate to use as either 10 Mbps, 100Mbps, or 1000Mbps. This command keyword must be following by the speed of the Ethernet connection, entered as an integer. ***Important:*** If the speed is manually configured, you must also configure the duplex mode. In addition, you must ensure that the network server configuration supports the speed and duplex configuration.
duplex	If the medium speed is manually configured, you must also configure the duplex mode through this parameter. Either **full** or **half** duplex mode can be implemented.
media	Specifies the SPIO Ethernet port media to use to communicate with the network server during boot. Select either **rj45**, for copper Ethernet, or the small form factor pluggable **sfp** optical gigabit Ethernet media type.

Step 3 Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring the Boot Network

Boot network parameters define the protocols and IP address information for SPIO interfaces used to reach the external network server that hosts the operating system software image file. To configure boot network parameters, make sure you are at the Exec mode prompt:

```
[local]host_name#
```

Step 1 Enter the Global Configuration mode by entering the following command:

```
configure
```

The following prompt appears:

```
[local]host_name(config)#
```

Step 2 Enter the following command:

```
boot networkconfig {dhcp|{{dhcp-static-fallback|static} ip address
spio24 <ip_address24> [spio25 <ip_address25>] netmask
<subnet_mask> [gateway <gw_ip_address>]}}
```

Keyword/Variable	Description
dhcp	Specifies the use of the Dynamic Host Control Protocol (DHCP) to automatically assign an IP address to the interface at startup. ***Important:*** If this option is selected, you will not have to configure IP address information for the SPIO interfaces, defined using the **boot interface** command, or any needed gateway.
dhcp-static-fallback	Specifies the use of the Dynamic Host Control Protocol (DHCP) to automatically assign an IP address to the SPIO interface, defined using the **boot interface** command, at startup. However, it allows the configuration of a fallback static IP address that can be used in case the DHCP server is unreachable.
static	Specifies that a static IP address will be configured for the SPIO's interface, defined using the **boot interface** command.
ip address	If either the **dhcp-static-fallback** or **static** options were used as the method by which the SPIO interface obtains an IP address, then these keywords specify the static address.
spio24 *ip_address24*	Specifies the IP address to use for the SPIO interface in slot 24. Enter the *ip_address24* variable as an IP address.
spio25 *ip_address25*	Specifies the IP address to use for the SPIO interface in slot 25. Enter the *ip_address25* variable as an IP address. If used, both interfaces will appear in the boot.sys file.
netmask	Enter the subnet mask, using dotted-decimal notation, that is used by each SPIO port.
gateway	If either **dhcp-static-fallback** or **static** options were chosen as the method by which the interface will receive an IP address, then this optional parameter specifies the IP address for the next-hop gateway (router, bridge, etc.) to use, if needed.

The following command configures the boot network to communicate using DHCP, with a static-fallback IP address for SPIO in slot 24 of 192.168.206.101 and a Class C netmask.

```
boot networkconfig dhcp-static-fallback ip address spio24
192.168.206.101 netmask 255.255.255.0
```

The next example uses static IP addresses for SPIOs in both slots 24 and 25, which can access the external network server through a gateway whose IP address is 135.212.10.2.

```
boot networkconfig static ip address spio24 192.168.206.101 spio25
192.168.206.102 netmask 255.255.255.0 gateway 135.212.10.2
```

Step 3 Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Boot Network Delay Time

An optional delay period, in seconds, can be configured for systems booting from a network. The purpose of this parameter is to allow time for external devices, such as switches, that use the Spanning Tree Protocol (STP) to determine the network route to a specified IP address.

To configure a boot network delay, enter the following command from the Global Configuration mode prompt.

```
boot delay <time>
```

Where *time* is entered as an integer, ranging from 1 to 300 seconds before attempting to contact the external network server. If your network uses STP, a typical delay time of 30 seconds should suffice.

Important: Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring a Boot Nameserver

To enter the hostname of the network server that hosts the operating system software image, first configure the IP address of the Domain Name Service (DNS) server, referred to as a name server, that can resolve the host name for the machine.

To configure a boot nameserver address, enter the following command from the Global Configuration mode prompt.

```
boot nameserver <ip_address>
```

Where *ip_address* is the IP address, entered in dotted-decimal notation, of the DNS server.

Important: Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Upgrading the Operating System Software

The following information is required prior to performing a software upgrade:

- Current operating system version
- New operating system version
- Whether the upgrade will be performed on-line or off-line

Identifying OS Release Version and Build Number

The operating system can be configured to provide services and perform pre-defined functions through issued commands from the CLI or through the Web Element Manager application.

The operating system software is delivered as a single binary file (**.bin** file extension) and is loaded as a single instance for the entire system. Each software image can be identified by its release version and its corresponding build number. The software version information can be viewed from the CLI by entering the **`show version`** command.

Software Upgrade Methods

There are two software upgrade methods used to add features, functionality, and correct known software defects. They are:

- On-Line Software Upgrade
- Off-line Software Upgrade

A brief overview accompanies each upgrade procedure.

Occasional software upgrades are required to add features and/or functionality, and to correct any previous defects.

On-Line Software Upgrade

This method is used to perform a software upgrade of the entire operating system.

Important: This method is not supported for the SGSN or for PDIF. Refer to the appropriate Administration Guide for upgrade information.

This method allows active sessions to be maintained until they are either self-terminated (subscriber ends session) or meet the optionally defined upgrade limit values.

This method upgrades all standby PACs/PSCs simultaneously, then upgrades any active cards simultaneously.

No new sessions will be accepted by the system during an on-line software upgrade.For PDSN and GGSN: All new session requests are blocked from entering the system through the use of an overload policy. Failure to configure this policy to redirect calls elsewhere can result in a significant service outage.

Caution: To minimize the risk of service outages, the on-line software upgrade should be performed during a planned maintenance window.

An on-line software upgrade is performed in five stages, where each stage is limited to performing only specific functions until the system is prepared to move to the next stage. Each stage is explained below.

CLI Verification and System Preparation

After initiating the upgrade command, before beginning Stage 1 of the on-line software upgrade process the system performs a series of checks and procedures. These include:

- Verifying that an open boot priority is available in the boot stack.
- Ensuring that the current local file system is synchronized.
- Creating the new boot stack entry using the new operating system image, boot priority, and configuration file information.
- Performing an SPC/SMC synchronization of the new local file system.
- Creating a temporary copy of the configuration that is currently running on the system. This configuration may or may not match the saved CLI configuration file that is named in the boot stack entry. This temporary copy is re-applied to the system during Stage 5 of the on-line software upgrade process.

If any errors are detected during this verification process, the on-line software upgrade is aborted and an error message is displayed.

Stage 1 - Soft Busy-out

For PDSN and GGSN: During this stage, all Session Manager tasks on the system are busied out and incoming session requests are redirected to other systems or rejected by the system, based on the configured overload policy for each service.

The system remains in this stage until either all current sessions are self-terminated by users or the configured session upgrade limits are reached. In the later case, when one of the two upgrade limits are reached, the system will automatically terminate all sessions that meet the time limit (maximum session life) or, when the usage limit (minimum # of sessions) on system is met and all sessions on the system are terminated.

Important: This is the only stage that the **`abort upgrade`** command may be used. Once Stage 2 is entered, the on-line software upgrade should not be cancelled unless an emergency exists. After Stage 1, the only way that an on-

line software upgrade can be terminated is to issue the **reload** command. This causes a system restart that could leave the system in an abnormal state, requiring manual intervention. Issuing the **reload** command should be avoided, and only used as a last resort.

Once all the calls on the system are terminated, the software upgrade enters Stage 2.

Stage 2 - Stand-alone Operation

In stage 2, the system switches from normal call operations, leaving only a minimal set of system-level tasks running on the PACs/PSCs to ensure that any errors are detected and, for PDSN and GGSN, that the re-directors used by the defined overload policy for each service remain in effect.

At this point, the SPCs/SMCs are fully operational, but each PAC/PSC in the system is running independently of the others, with no communications occurring between them. In this stage, the network processor units (NPUs) are placed into global bypass mode, wherein the redirector tasks are supported to deny any new session requests to access the system by redirecting them to other devices.

While in global bypass mode, Line Card (LC) ports will be limited to the following services:

- Respond to Ethernet ARP requests
- Respond to ICMP echo requests
- Session rejections or redirection

The following list defines LC features or services that will be unavailable:

- No AAA packets or logs will be sent for each session reject or redirect
- All other packets are discarded
- LC port counters will be unavailable
- Port redundancy operations, if configured, will not be operational
- All routing protocols, if enabled and configured, will be disabled
- Routing tables will remain fixed (no updates) throughout the upgrade
- PCF monitoring will be unavailable

Important: Once Stage 2 has begun, no CLI configuration mode commands, except **end** and **exit** (if this stage is entered while a management user is in a configuration mode) will be accepted by the system. Only non-configuration commands within the Exec mode, such as show commands may be executed. You can monitor the progress of the on-line software upgrade by entering the **show upgrade** command.

Once all of the PACs/PSCs are operating in stand-alone mode, the on-line software upgrade can proceed.

Stage 3 - Management Card Upgrade

During this stage, the system performs an SPC/SMC switchover, wherein all tasks running on the active SPC/SMC are transferred to the standby SPC/SMC, which then becomes active and takes control of the system.

The new standby SPC/SMC is then restarted and the new operating system software image is loaded onto that SPC/SMC. It is important to note that the full CLI configuration that was temporarily saved by the system is not loaded at this point. Instead, only minimal commands used to control the system are loaded.

Once this SPC/SMC is operational, another SPC/SMC switchover occurs and the second SPC/SMC is restarted, loading the new software version. During this period, since both SPCs/SMCs are effectively now running the new operating system software image, the system can continue to perform the on-line software upgrade process without waiting until the last SPC/SMC finishes booting up and is placed into its normal standby operational mode.

Stage 4 - Reboot All Packet Processing Cards

In this stage, the active SPC/SMC is aware of all system and card-level states and tasks. All PACs/PSCs that are in standby operational mode are restarted simultaneously, and after passing their POST diagnostics, their control processors (CPs) are loaded with the new operating system software image.

The remaining PACs/PSCs, which, for PDSN and GGSN, are enforcing the overload policies, preventing any new sessions from entering the system, are then migrated to the cards that are running the new operating system software. The overload policies and minimal system tasks continue running on the newly upgraded PACs/PSCs. The original active PACs/PSCs are then restarted, all at once, and upgraded to the new operating system software image.

Important: The system will only migrate as many active PACs/PSCs as there are standby PACs/PSCs. If this is not a 1:1 correlation, then the system will repeat this procedure of migrating - updating - migrating back until all normally active PACs/PSCs have been upgraded.

Once all of the cards have been upgraded and returned to their desired (normal) operating states, the system can proceed to the final stage of the on-line software upgrade procedure.

Stage 5 - Return System to Normal Operation

In this stage, all cards are running the new operating system software, but the full CLI configuration file that was created at the beginning of the upgrade has not yet been re-loaded and all network processor units (NPUs) are still operating in global bypass mode.

The system begins loading the full temporary CLI configuration file that was created at the beginning of the on-line software upgrade. This process can take over a minute to complete, dependent upon the size and complexity of the of the configuration file. As this process begins, the NPUs are programmed and all normal tasks are brought on-line, even though they are still in global bypass mode.

Once the configuration is fully loaded, returning the system to its pre-upgrade configuration, the system will switch the NPUs from global bypass mode. This cancels all redirection tasks configured by the overload policies, and the system can once again begin accepting new sessions.

System Requirements to Support the On-line Software Upgrade Method

A system requires a minimal amount of hardware to support this software upgrade method. The minimum required application cards are:

- Two SPCs/SMCs (one Active and one Standby)
- Two RCCs (required to support PAC/PSC migrations)
- Three PACs/PSCs (one must be a standby, but two standby cards are recommended)

If your system does not meet this minimal system requirement, then this method of software upgrade cannot be supported and you must use the Off-line Software Upgrade method, described later in this chapter.

Performing an On-line Software Upgrade

This procedure details how to successfully perform a software upgrade for operating system release version 3.5 and higher, using the on-line software upgrade method.

This procedure assumes that you have a CLI session established and are placing the new operating system image file onto the local file system. To begin, make sure you are at the Exec mode prompt:

[**local**]*host_name***#**

Step 1 Verify that there is enough free space on the device to accommodate the new operating system image file by entering the following command:

For ST16s:

```
directory {/flash|/pcmcia1|/pcmcia2}
```

For ASR 5000s:

```
directory {/flash|/pcmcia1|/hd-raid}
```

The following is an example of the type of directory information displayed:

```
-rwxrwxr-x 1 root root 7334 May 5 2003 startconfig.cfg
-rwxrwxr-x 1 root root 399 Jun 7 18:32 system.cfg
-rwxrwxr-x 1 root root 10667 May 14 16:24 testconfig.cfg
-rwxrwxr-x 1 root root 10667 Jun 1 11:21 testconfig_4.cfg
-rwxrwxr-x 1 root root 5926 Apr 7 2003 tworpcontext.cfg
-rwxrwxr-x 1 root root 15534 Aug 4 2003 test_vlan.cfg
-rwxrwxr-x 1 root root 2482 Nov 18 2002 gateway2.cfg
94844 /flash
Filesystem 1k-blocks Used Available Use% Mounted on
```

```
/dev/hda1 124778 94828 29950 76% /flash
```

Note the "Available" blocks in the last line of the display.

After displaying the directory information, it again returns to the root and the following prompt appears:

```
[local]host_name#
```

Step 2 View the boot stack entries and note the name and location (local device) of the CLI configuration file for the first entry (highest priority) by entering the following command:

```
show boot
```

Step 3 Verify that there are less than 10 boot stack entries in the boot.sys file and that a higher priority in the boot stack is available (i.e. that minimally there is no priority 1 entry in the boot stack). Refer to *Configuring the Boot Stack* for more information.

The system will automatically create a new boot stack entry for this software, using the <N-1> method, wherein the new entry will have a priority of one less than the previous entry (currently used).

Step 4 Using either an FTP client or the copy command, transfer the new operating system software image file to the location (network server or local SPC/SMC device) from where it will be loaded by the system.

For information on how to use the copy command, please reference the *Copying Files and Directories* section.

Caution: Whenever transferring a operating system software image file using the file transfer protocol (FTP), the FTP client must be configured to transfer the file using binary mode. Failure to use binary transfer mode will make the transferred operating system image file unusable.

Step 5 Back up the current CLI configuration file by entering the following command:

```
copy <from_url> <to_url> [-noconfirm]
```

For information on using the copy command, please see the *Copying Files and Directories* section.

The following command example creates a backup copy of a file called *general.cfg* located on the **/flash** device to a file called *general_3652.cfg*:

```
copy /flash/general.cfg /flash/general_3652.cfg
```

Step 6 Synchronize the local file systems on the SPCs/SMCs by entering one of the following commands:

For ST16s:

```
card spc synchronize filesystem all
```

For ASR 5000s:

```
card smc synchronize filesystem all
```

Step 7 Enter the Global Configuration mode by entering the following command:

```
configure
```

The following prompt appears:

```
[local]host_name(config)#
```

Step 8 Configure upgrade session limits by entering the following command:

```
upgrade limit time <session_life> usage <session_num>
```

The system supports thresholds for both session time and number of sessions. These parameters are used by the system to identify when it may execute the software upgrade process.

Keyword/Variable	Description
upgrade limit	This command issued with no keywords sets all parameters to their defaults.
time *session_life*	Default: 120 Defines the maximum number of minutes that a session may exist on the system, undergoing a software upgrade or installation, before it is terminated by the system. As individual user sessions reach this lifetime limit, the system terminates the individual session(s). *session_life* must be an integer ranging from 1 through 1440.
usage *session_num*	Default: 100 This keyword defines a low threshold limit of sessions running either on a PAC/PSC or system-wide. When a software upgrade is invoked, this parameter applies to the entire system. When the threshold is crossed (when the number of sessions on the PAC/PSC or system is less than this value), the remaining sessions on the PAC/PSC or system are terminated allowing the upgrade to begin. The remaining sessions on the PAC/PSC or system are terminated regardless of their session life. *session_num* must be an integer from 0 through 6000.

Step a Enter the Context Configuration mode by entering the following command:

```
context <context_name>
```

The following prompt appears:

```
[<context_name>]host_name(config-ctx)#
```

Step b Enter the ASN GW service configuration mode for the service to be configured by entering the following command:

```
asngw-service <service_name>
```

The following prompt appears:

```
[<context_name>]host_name(config-asngw-service)#
```

Step c Configure the overload policy for this service by entering the following command:

```
policy overload {drop|reject}
```

Step d *Optional.* Configure the overload policy for another configured ASN GW service.

Keyword/Variable	Description
drop	Default: disabled Specifies that the system is to drop incoming packets containing new session requests.
reject	Default: enabled Specifies that the system processes new session request messages and responds with a reject message.

Step 9 For PDSN and HA services, configure an overload policy for each service that redirects new session requests to other devices or rejects them as given procedure below.

Step a Enter the Context Configuration mode by entering the following command:

```
context <context_name>
```

The following prompt appears:

```
[<context_name>]host_name(config-ctx)#
```

Step b Enter the Service Configuration mode for the service to be configured by entering the following command:

```
{pdsn-service|ha-service} <service_name>
```

The following prompt appears:

```
[<context_name>]host_name(config-<service_type>-service)#
```

Step c Configure the overload policy for this service by entering the following command:

```
policy {overload {redirect <address> [weight <weight_num>]
[<address2> [weight <weight_num>]...<address16> [weight
<weight_num>] ] | reject [use-reject-code insufficient-resources]}
| service-option enforce}
```

Keyword/Variable	Description
redirect *<ip_address>*	Enables a redirect policy for overloading conditions. When a redirect policy is invoked, the PDSN service rejects new sessions with an A11 Registration Reply Code of 88H (unknown PDSN address) and provides the IP address of an alternate PDSN. This command can be issued multiple times. *address*: The IP address of an alternate PDSN expressed in IP v4. Up to 16 IP addresses can be specified either in one command or by issuing the redirect command multiple times. If you try to add more than 16 IP addresses to the redirect policy the CLI issues an error message. If you specify an IP address and weight that already exists in the redirect policy the new values override the existing values.
weight *<weight_num>*	When multiple addresses are specified, they are selected in a weighted round-robin scheme. Addresses with higher weights are more likely to be selected when redirecting traffic. If a weight is not specified the entry is automatically assigned a weight of 1. *weight_num* must be an integer from 1 through 10.
reject	Specifies that the service should reject all incoming session requests, returning a result code (81H) indicating "**Registration Denied - Administratively Prohibited**" to the requestor.
use-reject-code insufficient-resources	*Optional:* This optional keyword may be used in conjunction with a reject overload policy for either PDSN or HA services. The result of this command is that a result code (82H) indicating "**Registration Denied - Insufficient Resources**" is returned to the requestor.
service-option enforce	If enabled, the system will only allow calls that contain the service option(s) configured using the service option command. If disabled, the system will not check for the service option and allow calls regardless of the service option they support.

Step d Repeat *step c* to configure the overload policy for another configured service.

Step 10 Return to the Exec mode prompt by entering the following command:

```
end
```

The following prompt appears:

```
[local]host_name#
```

Caution: Once the software upgrade process has started, any failure that results in the reboot of the system prior to the upgrading of both SPCs/SMCs may result in unexpected behavior by the system that requires manual intervention to correct.

Step 11 Save your configuration as described in *Verifying and Saving Your Configuration*.

Step 12 Begin the on-line software upgrade by entering the following command:

```
upgrade online <image_url> config <cfg_url> [-noconfirm]
```

Keyword/Variable	Description
-noconfirm	Disables the "**Are you sure? [Yes | No]**" confirmation prompt asked before executing the command.

The SPCs/SMCs within the system load the new operating system image and the local file system is synchronized. The system then updates all standby PACs/PSCs. Next, it begins to update each active PAC/PSC, one at a time. The system monitors all sessions being processed by active PACs/PSCs. When all sessions facilitated by a specific Session Manager task are either self-terminated or automatically terminated based on the thresholds configured in step 8, the system migrates the PAC/PSC in active mode to standby mode. The new standby PAC/PSC is upgraded and rebooted. Once booted, the card is placed back into service as an active PAC/PSC.

Step 13 *Optional:* To view the status of an on-line software process, enter the following command from the Exec mode prompt:

```
show upgrade
```

This command displays the status of the on-going on-line software upgrade.

Once all PACs/PSCs have been upgraded, the full configuration file is loaded, the NPUs are taken out of global bypass mode, and the system is returned to normal operation. When the on-line software upgrade has been completed, all sessions on the system will be new and all system statistics will have been reset.

Upon completion of the software upgrade, the system will automatically begin accepting new sessions, using the pre-existing configuration that was running. All system statistical counters will have been reset to zero.

Aborting an On-line Software Upgrade

Abort the on-line software upgrade process by entering the following command:

```
abort upgrade [-noconfirm]
```

Important: The abort upgrade command can only be used during Stage 1 (busy-out) of an on-line software upgrade.

Restoring the Previous (Pre-online Upgrade) Software Image

If for some reason you need to restore the system to the software image that was running before the online upgrade process, perform the *On-Line Software Upgrade* again and specify the locations of the original software image and configuration files.

Performing an Off-line Software Upgrade

An off-line software upgrade can be performed for any system, upgrading from any version of operating system software to any version, regardless of version number. This process is considered off-line because while many of the steps can be performed while the system is currently supporting sessions, the last step of this process requires a reboot to actually apply the software upgrade.

Optional for PDSN: If you want to use the IP Pool Sharing Protocol during your upgrade, refer to Configuring IPSP Before the Software Upgrade in the IP Pool Sharing Protocol chapter of the System Enhanced Feature Configuration Guide.

This procedure assumes that you have a CLI session established and are placing the new operating system image file onto the local file system. To begin, make sure you are at the Exec mode prompt:

```
[local]host_name#
```

Step 1 Verify that there is enough free space on the device to accommodate the new operating system image file by entering the following command:

For ST16s:

```
directory { /flash | /pcmcia1 | /pcmcia2 }
```

For ASR 5000s:

```
directory { /flash | /pcmcia1 | /hd-raid }
```

The following is an example of the type of directory information displayed:

```
-rwxrwxr-x 1 root root 7334 May 5 2003 startconfig.cfg
-rwxrwxr-x 1 root root 399 Jun 7 18:32 system.cfg
-rwxrwxr-x 1 root root 10667 May 14 16:24 testconfig.cfg
-rwxrwxr-x 1 root root 10667 Jun 1 11:21 testconfig_4.cfg
-rwxrwxr-x 1 root root 5926 Apr 7 2003 tworpcontext.cfg
-rwxrwxr-x 1 root root 15534 Aug 4 2003 test_vlan.cfg
-rwxrwxr-x 1 root root 2482 Nov 18 2002 gateway2.cfg
94844 /flash
Filesystem 1k-blocks Used Available Use% Mounted on
/dev/hda1 124778 94828 29950 76% /flash
```

Note the “Available” blocks in the last line of the display.

After displaying the directory information, it again returns to the root and the following prompt appears:

```
[local]host_name#
```

Step 2 Transfer the new operating system image file to the local device, if needed, by using one of the following:

Step a Copy the file from an external device or other local device on the SPC/SMC by entering the following command:

```
copy <from_url> <to_url> [-noconfirm]
```

For information on using the copy command, please see the *Copying Files and Directories* section.

Step b Transfer the file to the local device using an FTP client with access to the system.

Caution: Whenever transferring a operating system software image file using the file transfer protocol (FTP), the FTP client must be configured to transfer the file using binary mode. Failure to use binary transfer mode will make the transferred operating system image file unusable.

Step 3 Back up the current CLI configuration file by entering the following command:

```
copy <from_url> <to_url> [-noconfirm]
```

This creates a mirror-image of the CLI configuration file linked to the operating system defined in the current boot stack entry.

The following command example creates a backup copy of a file called `general.cfg` located on the **/flash** device to a file called `general_3652.cfg`:

```
copy /flash/general.cfg /flash/general_3652.cfg
```

Step 4 Modify the boot stack entry for the current file group to reflect the filename change for the current entry by entering the following command:

```
configure

    boot system priority <number> image <image_url> config
<cfg_url>
```

Important: The maximum number of boot stack entries that can be contained in the boot.sys file is 10. If there are already 10 entries in the boot stack, then you must delete at least one of these entries before proceeding. Refer to Configuring the Boot Stack for more information.

For information on using the **boot system priority** command, refer to the *Adding a New Boot Stack Entry* section.

Step 5 Create a new boot stack entry for the new file group, consisting of the new operating system image file and the currently used CLI configuration file by entering the following command:

```
boot system priority <number> image <image_url> config <cfg_url>
```

Assign the next highest priority to this entry, by using the <N-1> method, wherein you assign a priority number that is one number less than your current highest priority.

For information on using the **boot system priority** command, please see the *Adding a New Boot Stack Entry* section.

Step 6 Synchronize the local file systems on the SPCs/SMCs by entering one of the following commands:

For SPCs:

```
card spc synchronize filesystem {/flash|/pcmcia1|/pcmcia2|all}
[checkonly] [reverse]} [-noconfirm]
```

For SMCs:

```
card smc synchronize filesystem {/flash|all} [checkonly]
[reverse]} [-noconfirm ]
```

Keyword/Variable	Description
/flash	Synchronizes only the CompactFlash file system on the standby SPC/SMC.
/pcmcia1	Synchronizes only the file system of the PCMCIA card installed in the PCMCIA 1 slot on the standby SPC/SMC.
/pcmcia2	Synchronizes only the file system of the PCMCIA card installed in the PCMCIA 2 slot on the standby SPC. ***Important:*** This option is not supported for use with SMCs.
all	Specifies that filesystems on all available matching local devices (/flash, /pcmcia1, /pcmcia2, and/or /hd-raid) be synchronized. ***Important:*** Only filesystems on matching local devices will be synchronized. For example, if the active SPC/SMC contains two local devices (/flash and /pcmcia1) and the standby SPC/SMC contains only one local device (/flash), then synchronization would only occur on the matching local device (i.e. /flash).
checkonly	Displays a list of files that would be synchronized, without executing any synchronization actions.
reverse	Performs the specified operation on the standby SPC/SMC.

Keyword/Variable	Description
-noconfirm	This keyword disables the "**Are you sure? [Yes | No]**" confirmation prompt, asked before executing the command

Step 7 Configure a newcall policy from the Exec mode. Newcall policies are created on a per-service basis and can be routed to another service running on the same device if no external device running services is available:

```
newcall policy {fa-service|ha service | ggsn-service | mme-service
| pdsn-service|asngw-service|asnpc-service|lns-service} {all |
name <service_name>} reject
```

```
newcall policy {ha-service|pdsn-service} {all|name <service_name>}
{redirect <target_ip_address> [weight <weight_num>]
[<target_ipaddress2> [weight <weight_num>]...<target_ip_address16>
[weight <weight_num>]] |reject}
```

Keyword/Variable	Description
name	Specifies a single instance of a service type or an APN to apply the newcall policy to. *service_name* is the name of a service that was previously configured. It can consist of up to 63 alphanumeric characters and is case sensitive. *apn_name* is the name of an APN that was previously configured. It can consist of up to 63 alphanumeric characters and is case sensitive. ***Important:*** To apply the newcall policy to a subset of all of the configured services of a specific type, re-issue the command for each individual service name desired.
redirect	Configures the busy-out action. When a redirect policy is invoked, the service rejects new sessions and provides the IP address of an alternate destination. This command can be issued multiple times. *address:* The IP address of an alternate destination expressed in IP v4. Up to 16 IP addresses can be specified either in one command or by issuing the redirect command multiple times. If you try to add more than 16 IP addresses to the redirect policy the CLI issues an error message. If you specify an IP address and weight that already exists in the redirect policy the new values override the existing values Depending on the type of service that the policy is applied to, a reason code is reported as part of the reply to indicate the rejection reason.
weight *weight_num*	When multiple addresses are specified, they are selected in a weighted round-robin scheme. Addresses with higher weights are more likely to be selected when redirecting traffic. If a weight is not specified the entry is automatically assigned a weight of 1. *weight_num* must be an integer from 1 through 10.
reject	Specifies that the policy will reject new incoming session requests.

Step 8 *Optional:* Configure a newcall policy for each additional service type.

Step 9 *Optional:* Configure a "Message of the Day" banner informing other management users that the system will be rebooted by entering the following command from the Global Configuration mode prompt.

```
banner motd "banner_text"
```

banner_text is the message that you would like to be displayed and can be up to 2048 alpha and/or numeric characters. Note that *banner_text* must begin with and end in quotation marks (“ “). For more information in entering CLI banner information, please see the *CLI Command Reference* documentation.

The banner is displayed when an administrative user logs onto the CLI.

Step 10 Reboot by entering the following command:

```
reload [-noconfirm]
```

As the system reboots, it loads the new operating system software image and its corresponding CLI configuration file using the new boot stack entry configured earlier.

Important: After the system reboots, establish a CLI session and enter the command to verify that the active software version is correct.

Step 11 *Optional for PDSN:* If you are using the IP Pool Sharing Protocol during your upgrade, wait until the system is completely rebooted, then go to *Configuring IPSP After the Software Upgrade* in the *IP Pool Sharing Protocol* chapter of the *System Enhanced Feature Configuration Guide*.

Restoring the Previous Software Image

If for some reason you need to undo the upgrade, perform the upgrade again except:

- Specify the locations of the upgrade software image and configuration files

then

- Specify the locations of the original software image and configuration files

Managing License Keys

License keys define capacity limits (number of allowed subscriber sessions) and available features on your system. Adding new license keys allows you to increase capacity and add new features as your subscriber base grows.

New System License Keys

New systems are delivered with no license keys installed. In most cases, you receive the license key in electronic format (usually through email).

When a system boots with no license key installed a set of default limited session use and feature licenses is installed. The following Exec Mode command lists the license information:

```
show license information
```

Important: With no license key installed, the session use licenses for PDSN, HA, GGSN, and L2TP LNS are limited to 10,000 sessions.

SPCs/SMCs are shipped with a CompactFlash card installed. A single license key is generated using the serial numbers from the CompactFlash cards. If two SPCs/SMCs are installed, the license key is generated from the serial numbers of both CompactFlash cards. This allows the license to be distributed across both SPCs/SMCs, ensuring that licensed capacity and features remain available during a switchover event.

Installing New License Keys

Use the instructions below to install a new license key.

Cutting and Pasting the Key

If you have a copy of the license, use the following configuration to cut and paste just the licence key part:

Step 1 From the Exec mode, enter the following:

```
configure
   license key license
   exit
```

license is the license key string.

The license can be between 1 and 1023 alpha-numeric characters and is case sensitive.

Copy the license key as shown in the example below, including the "\. Please note: this is not a functional license.

```
"\

VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-
1|C2S=\STCB21M82003R80411A4|DOI=0000000000|DOE=00000000|ISS=1|NUM=
13459|0000000000000|LSP=000000|LSH=000000|LSG=500000|LSL=500000\|F
IS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|FMG=Y|FCR=Y|FSR=Y|FPM=Y|FID=Y|SIG=MCw
CF\Esnq6Bs/XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd6388jHsHD99sg36SG267gshs
sja77

end
```

Step 2 Verify that the license key just entered was accepted by entering the following command at the Exec mode prompt:

```
show license key
```

The new license key should be displayed. If it is not, return to the Global configuration mode and re-enter the key using the **license key** command.

Step 3 Verify that the license key enabled the correct functionality by entering the following command:

```
show license information
```

All license keys and the new session capacity or functionality enabled should be listed. If the functionality or session capacity enabled by the new key is incorrect, please contact your service representative.

Step 4 Save your configuration as described in *Verifying and Saving Your Configuration*.

Caution: Failure to save the new license key configuration in the current CLI configuration file will result in the loss of any of the new features enabled by the license key once the system is reloaded.

Adding License Keys to Configuration Files

License keys can be added to a new or existing configuration file.

Important: License key information is maintained as part of the CLI configuration. Each time a key is installed or updated, you must re-save the configuration file.

Step 1 Open the configuration file to which the new license key commands are to be copied.

Step 2 Copy the license as shown in the example, including the "\. Please note: this is not a functional license.

```
"\

VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-
1|C2S=\STCB21M82003R80411A4|DOI=0000000000|DOE=00000000|ISS=1|NUM=
13459|0000000000000|LSP=000000|LSH=000000|LSG=500000|LSL=500000\|F
IS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|FMG=Y|FCR=Y|FSR=Y|FPM=Y|FID=Y|SIG=MCw
CF\Esnq6Bs/XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd6388jHsHD99sg36SG267gshs
sja77

end
```

Step 3 Paste the license key into the configuration

> ***Important:*** Paste the license key information at the beginning of the configuration file to ensure the system has the expected capacity and features before it configures contexts.

Step 4 Save your configuration as described in *Verifying and Saving Your Configuration*.

License Expiration Behavior

When a license expires, there is a built-in grace period of 30 days that allows normal use of the licensed session use and feature use licenses. This allows you to obtain a new license without any interruption of service.

The following Exec mode command lists the license information including the date the grace period is set to expire:

```
show license information
```

The following example shows the license information for a system with an expired license key installed. The boldfaced text shows the grace period information:

```
Key Information (installed key):

 Comment                <Host Name>

 CF Device 1 Model: "SanDiskSDCFB-512"

 Serial Number: "101904J1204Q2810"

 CF Device 2 Model: "SanDiskSDCFB-512"

 Serial Number: "003507E2004H0627"

 Date of Issue Thursday June 09 16:03:04 EDT 2005

 Issued By              <Vendor Name>

 Key Number 17240

 Enabled Features:
```

```
Part Number Quantity Feature
----------- -------- -----------------------
xxx-xx-xxxx         23 PDSN (10K)
xxx-xx-xxxx          8 PDSN (1K)
[none] -  FA
xxx-xx-xxxx         22 HA (10K)
xxx-xx-xxxx          8 HA (1K)
[none] -  IPv4 Routing Protocols
xxx-xx-xxxx          - IPSec
xxx-xx-xxxx          - Prepaid Accounting
xxx-xx-xxxx          - L2TP LAC (PDSN)
xxx-xx-xxxx          - L2TP LAC (HA)
xxx-xx-xxxx          1 L2TP LNS (1K)
xxx-xx-xxxx          - PDSN Closed RP
  .....
  .....
  xxx-xx-xxxx          - Destination Based Accounting
xxx-xx-xxxx          - Layer 2 Traffic Management
xxx-xx-xxxx          - Dynamic Mobile IP Key Update
Session Limits:
Sessions Session Type
-------- -----------------------
238000  PDSN
228000 HA
1000 L2TP LNS
Status:
CF Device 1 Does not match either SPC
CF Device 2 Does not match either SPC
License Status Not Valid [In Grace Period]
```

```
Grace Period Ends Thursday March 14 15:56:13 EDT 2009
```

Requesting License Keys

License keys for the system can be obtained through your local sales or customer support representative. Specific information is required before a license key may be generated:

- Sales Order or Purchase Order information
- Desired session capacity
- Desired functionality
- CompactFlash detail

To obtain the model and serial number of a CompactFlash card, enter the following command at the Exec mode prompt:

show card information *<slot#>*

Where *slot#* is either 8 or 9, depending on the chassis card slot where the SPC/SMC is installed.

The following example provides the output for an SPC in slot 8. The compact flash information is in bold text.

```
Card 8:
 Slot Type : SPC
 Card Type : Switch Processor Card
 Operational State : Active
 Last State Change : Tuesday July 27 09:57:48 EDT 2004
 Administrative State : Enabled
 Card Lock : Locked
 Reboot Pending : No
 Card Usable : Yes
 Single Point of Failure : Yes, needs a Switch Processor Card in slot 9
 Attachment : 24 (Switch Processor I/O Card)
 Attachment : 25 (Switch Processor I/O Card)
 Temperature : 38 C (limit 84 C)
 Voltages : Good
 Card LEDs : Run/Fail: Green | Active: Green | Standby: Off
```

```
 System LEDs : Status: Green | Service: Off
Compact Flash : Present
 Type : 122M disk
 Model : TOSHIBATHNCF128MBA
 Serial Number : STCB21M82003R80411A4
 PCMCIA 1 : Present
 Type : 122M disk
 Model : SanDiskSDCFB-128
 Serial Number : 12090110228
 PCMCIA 2 : Not Present
 CPU 0 : Diags/Kernel Running, Tasks Running
```

Viewing License Information

To see the license detail, enter the following command from the Exec mode:

show license information

The following example displays the output of this command for the same system with a valid license key installed.

```
Key Information (installed key):
 Comment                <Host Name>
 CF Device 1 Model: "SanDiskSDCFB-512"
 Serial Number: "115212D1904T0314"
 CF Device 2 Model: "SanDiskSDCFB-512"
 Serial Number: "115206D1904S5951"
 Date of Issue Thursday May 12 14:35:50 EDT 2005
 Issued By              <Vendor Name>
 Key Number 17120
 Enabled Features:
 Part Number Quantity Feature
 ----------- -------- -----------------------
```

```
xxx-xx-xxxx         15 PDSN (10K)
[none] - FA
[none] - IPv4 Routing Protocols
xxx-xx-xxxx          - IPSec
xxx-xx-xxxx          - L2TP LAC (PDSN)
xxx-xx-xxxx          1 L2TP LNS (10K)
xxx-xx-xxxx          6 L2TP LNS (1K)
xxx-xx-xxxx          - PDSN Closed RP
xxx-xx-xxxx          - Session Recovery (PDSN)
       [none]         - Session Recovery (HA)
xxx-xx-xxxx          - Lawful Intercept
  xxx-xx-xxxx          - PCF Monitoring
xxx-xx-xxxx          - Layer 2 Traffic Management
Session Limits:
Sessions Session Type
-------- -----------------------
150000  PDSN
16000 L2TP LNS
Status:
CF Device 1 Does not match either SPC
CF Device 2 Does not match either SPC
License Status Good (Not Redundant)
```

Deleting a License Key

Use the procedure below to delete the session and feature use license key from a configuration. You must be a security administrator or administrator.

```
configure
   no license key
```

```
exit
show license key
```

The output of this command should display:

```
No license key installed
```

Management Card Replacement and License Keys

In the event that an individual SPC/SMC is replaced, the CompactFlash card on the new SPC/SMC must be exchanged with the CompactFlash from the original SPC/SMC because the license key was generated based on the serial number of the CompactFlash card associated with the original SPC/SMC.

Exchanging the two CompactFlash card modules ensures that license redundancy is maintained, as the license key will continue to match both CompactFlash serial numbers on both SPCs/SMCs.

Important: Failure to provide license key redundancy can result in the loss of session capacity and enhanced features should a failover or manual switchover occur.

Instructions for the removal and installation of the CompactFlash on SPCs/SMCs can be found in the *Hardware Installation Guide*.

Managing Local-User Administrative Accounts

Unlike context-level administrative accounts which are configured via a configuration file, information for local-user administrative accounts is maintained in a separate file on the CompactFlash and managed through the software's Shared Configuration Task (SCT). Because local-user accounts were designed to be compliant with ANSI T1.276-2003, the system provides a number of mechanisms for managing these types of administrative user accounts.

Configuring Local-User Password Properties

Local-user account password properties are configured globally and apply to all local-user accounts. The system supports the configuration of the following password properties:

- **Complexity:** Password complexity can be forced to be compliant with ANSI T1.276-2003.
- **History length:** How many previous password versions should be tracked by the system.
- **Maximum age:** How long a user can use the same password.
- **Minimum number of characters to change:** How many characters must be changed in the password during a reset.
- **Minimum change interval:** How often a user can change their password.
- **Minimum length:** The minimum number of characters a valid password must contain.

Refer to the `local-user password` command in the *Global Configuration Mode* chapter of the *Command Line Interface Reference* for details on each of the above parameters.

Configuring Local-User Account Management Properties

Local-user account management includes configuring account lockouts and user suspensions.

Local-User Account Lockouts

Local-user accounts can be administratively locked for the following reasons:

- **Login failures:** The configured maximum login failure threshold has been reached. Refer to the `local-user max-failed-logins` command in the *Global Configuration Mode* chapter of the *Command Line Interface Reference* for details
- **Password Aging:** The configured maximum password age has been reached. Refer to the `local-user password` command in the *Global Configuration Mode* chapter of the *Command Line Interface Reference* for details.

Accounts that are locked out are inaccessible to the user until either the configured lockout time is reached (refer to the **local-user lockout-time** command in the *Global Configuration Mode* chapter of the *Command Line Interface Reference*) or a security administrator clears the lockout (refer to the **clear local-user** command in the *Exec Mode* chapter of the *Command Line Interface Reference*).

Important: Local-user administrative user accounts could be configured to enforce or reject lockouts. Refer to the **local-user username** command in the *Global Configuration Mode* chapter of the *Command Line Interface Reference* for details.

Local-User Account Suspensions

Local-user accounts can be suspended as follows:

```
configure
   suspend local-user <name>
```

A suspension can be removed by entering:

```
configure
   no suspend local-user <name>
```

Changing Local-User Passwords

Local-user administrative users can change their passwords using the **password change** command in the Exec mode. Users are prompted to enter their current and new passwords.

Security administrators can reset passwords for local-users by entering the following command from the root prompt in the Exec mode:

```
password change username <name>
```

name is the name of the local-user account for which the password is to be changed. When a security administrator resets a local-user's password, the system prompts the user to change their password the next time they login.

All new passwords must adhere to the password properties configured for the system.

Chapter 9
Monitoring the System

This chapter provides information for monitoring system status and performance using the **show** commands found in the Command Line Interface (CLI). These command have many related keywords that allow them to provide useful information on all aspects of the system ranging from current software configuration through call activity and status.

The selection of keywords described in this chapter is intended to provided the most useful and in-depth information for monitoring the system. For additional information on these and other **show** command keywords, refer to the *Command Line Interface Reference*.

SNMP Notifications

In addition to the CLI, the system supports Simple Network Management Protocol (SNMP) notifications that indicate status and alarm conditions. Refer to the *SNMP MIB Reference* for a detailed listing of these notifications.

Monitoring System Status and Performance

This section contains commands used to monitor the status of tasks, managers, applications and other software components in the system. Output descriptions for most of the commands are located in the *Statistics and Counters Reference*.

To do this:	Enter this command:
View Administrative Information	
Display Current Administrative User Access	
View a list of all administrative users currently logged on the system	`show administrators`
View the context in which the administrative user is working, the IP address from which the administrative user is accessing the CLI, and a system generated ID number	`show administrators session id`
View information pertaining to local-user administrative accounts configured for the system	`show local-user verbose`
View statistics for local-user administrative accounts	`show local-user statistics verbose`
View information pertaining to your CLI session	`show cli`
Determining System Uptime	
View system uptime (time since last reboot)	`show system uptime`
View NTP Server Status	
View NTP servers status	`show ntp status`
View System Resources	
View all system resources such as CPU resources and number of managers created	`show resources [ cpu ]`
View System Alarms	
View information about all currently outstanding alarms	`show alarm outstanding all verbose`
View system alarm statistics	`show alarm statistics`
View Congestion-Control Statistics	
View Congestion-Control Statistics	`show congestion-control statistics`
View Remote Management Statistics	
Display SNMP Notification Statistics	
View SNMP notification statistics	`show snmp notifies`
Display SNMP Access Statistics	
View SNMP access statistics	`show snmp accesses`
Display SNMP Trap History	

To do this:	Enter this command:
View SNMP trap history	**show snmp trap history**
Display SNMP Trap Statistics	
View SNMP Trap Statistics	**show snmp trap statistics**
Display ORBEM Information	
View ORBEM client status	**show orbem client id**
View ORBEM session information	**show orbem session table**
View individual ORBEM sessions	**show orbem session id orbem**
View ORBEM status information	**show orbem status**
View Port Counters	
Display Port Datalink Counters	
View datalink counters for a specific port	**show port datalink counters** *slot#/port#*
Display Port Network Processor Unit (NPU) Counters	
View NPU counters for a specific port	**show port npu counters** *slot#port#*

Important: The commands or keywords/variables that are available are dependent on platform type, product version, and installed license(s). Some commands have different outputs depending on the platform type.

Clearing Statistics and Counters

It may be necessary to periodically clear statistics and counters in order to gather new information. The system provides the ability to clear statistics and counters based on their grouping (PPP, MIPHA, MIPFA, etc.).

Statistics and counters can be cleared using the CLI **`clear`** command. Refer to the *Command Line Interface Reference* for detailed information on using this command.

Chapter 10
Monitoring Hardware Status

This chapter describes how to use the command line interface (CLI) **show** commands to monitor system status and performance. These command have related keywords that you can use to get information on all aspects of the system, ranging from current software configuration through call activity and status.

The selection of keywords described in this chapter provides useful and in-depth information for monitoring the hardware. For additional information on these and other **show** command keywords, refer to the *Command Line Interface Reference*.

SNMP Notifications

In addition to the CLI, the system supports Simple Network Management Protocol (SNMP) notifications that indicate status and alarm conditions. Refer to the *SNMP MIB Reference Guide* for a detailed list.

Monitoring Hardware Status

Use the commands contained in this section to monitor the status of the hardware components in the chassis. For output descriptions for most of the commands, refer to the Counters and Statistics Reference.

Important: The commands or keywords and variables are dependent on platform type, product version, and installed license(s). Some commands produce different outputs, depending on the platform type.

Chapter 11
Configuring and Maintaining Bulk Statistics

This chapter provides configuration information for:

- Configuring Communication With the Collection Server
- Viewing Collected Bulk Statistics Data
- Manually Gathering and Transferring Bulk Statistics
- Clearing Bulk Statistics Counters and Information
- Bulk Statistics Event Log Messages

Configuring Communication With the Collection Server

Two configuration methods are available for defining how bulk statistics are collected and managed. A “standard” configuration allows the system to automatically assign a number to the bulk statistic file. Optionally, a number can be specified by an administrator in the optional configuration method. Command details and descriptions of keywords and variables for commands in this chapter are located in the *Bulk Statistics Configuration Mode Commands* chapter and the *Bulk Statistics File Configuration Mode Commands* chapter located in the *Command Line Interface Reference*.

Configuring Standard Settings

The configuration example in this section defines basic operation of the bulk statistics feature. Use the following example configuration to set up the system to communicate with the statistic collection server:

Important: Some CPU statistics from the Card schema are now located in the System schema. These are tabled in the Unsupported Management Appendix along with a list of other unsupported bulk statistics. Supported bulk statistics are in the *Statistics Counters Reference*.

```
configure
   bulkstats mode
      schema <name> format <format_string>
      sample-interval <time_interval>
      transfer-interval <xmit_time_interval>
      limit <mem_limit>
      exit
   bulkstats collection
   end
```

Configuring Optional Settings

This section describes optional commands that can be used within the bulk statistics configuration mode. Specifically, bulk statistic “files” under which to group the bulk statistic configuration are configured using commands in this section. “Files” are used to group bulk statistic schema, delivery options, and receiver configuration. Because multiple “files” can be configured, this functionality provides greater flexibility in that it allows you to configure different schemas to go to different receivers.

```
configure
   bulkstats mode
      file <number>
         receiver <ip_address> { primary | secondary } [ mechanism { { {
ftp | sftp } login <user_name> [ encrypted ] password <pwd> } | tftp } }
] }
         receiver mode { redundant | secondary-on-failure }
         remotefile format <naming_convention> [ both-receivers |
primary-receiver | secondary-receiver ]
         header format <header_format>
         footer format <footer_format>
         exit
      <name> schema format <format_string>
      sample-interval <time_interval>
      transfer-interval <xmit_time_interval>
      limit <mem_limit>
      exit
   bulkstats collection
   end
```

Configuring Bulk Statistic Schemas

In each configuration example described in *Configuring Standard Settings* and *Configuring Optional Settings*, the command "`<name>` schema format `<format_string>`" is the primary command used to configure the type of schema and the statistics collected. Refer to the *Bulk Statistics Configuration Mode Commands* chapter and the *Bulk Statistics File Configuration Mode Commands* chapter located in the *Command Line Interface Reference* for more information regarding supported schemas, available statistics, and proper command syntax.

Verifying Your Configuration

After configuring support for bulk statistics on the system, you can check your settings prior to saving them.

Follow the instructions in this section to verify your bulk statistic settings. These instructions assume that you are at the root prompt for the Exec mode.

Check your collection server communication and schema settings by entering the following command:

```
show bulkstats schema
```

The following is an example command output:

```
Bulk Statistics Server Configuration:

 Server State: Enabled

 File Limit: 1000 KB

 Sample Interval: 1 minutes (0D 0H 1M)

 Transfer Interval: 5 minutes (0D 0H 5M)

 Collection Mode: Cumulative

 Receiver Mode: Secondary-on-failure

 Remote File Format:
/users/ems/server/data/chicago/bulkstat%date%%time%.txt

 File Header: "CHI_test %time%"

 File Footer: ""

 Local File Storage: None

Bulk Statistics Server Statistics:

 Records awaiting transmission: 114

 Bytes awaiting transmission: 8092

 Total records collected: 59926

 Total bytes collected: 4190178

 Total records transmitted: 59812

 Total bytes transmitted: 4188512

 Total records discarded: 0

 Total bytes discarded: 0

 Last collection time required: 2 second(s)

 Last transfer time required: 0 second(s)

 Last successful transfer: Wednesday July 28 12:14:30 EDT 2004
```

```
 Last successful tx recs: 190
 Last successful tx bytes: 13507
 Last attempted transfer: Wednesday April 20 12:14:30 EDT 2009
Bulkstats Receivers:
 Primary: 192.168.0.100 using FTP with username administrator

Type            Name            Format
---------       ---------       ----------------------------
port            portstats       %bcast_inpackets% - %bcast_outpackets%
```

Saving Your Configuration

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Viewing Collected Bulk Statistics Data

The system provides a mechanism for viewing data that has been collected but has not been transferred. This data is referred to as “pending data”.

View pending bulk statistics data per schema by entering the following:

```
show bulkstats data
```

The above command also shows the statistics of remote files, if configured in the *Configuring Optional Settings* section of *Configuring Communication With the Collection Server* in this chapter.

The following is a sample output:

```
Bulk Statistics Server Statistics:
 Records awaiting transmission: 1800
 Bytes awaiting transmission: 163687
 Total records collected: 1800
 Total bytes collected: 163687
 Total records transmitted: 0
 Total bytes transmitted: 0
 Total records discarded: 0
 Total bytes discarded: 0
 Last collection time required: 2 second(s)
 Last transfer time required: 0 second(s)
 No successful data transfers
 Last attempted transfer: Tuesday February 14 15:12:30 EST 2006
 File 1
 Remote File Format: /users/server/data/bulkstat%date%%time%.txt
 File Header: "Format 4.5.3.0"
 File Footer: ""
 Bulkstats Receivers:
 Primary: 192.168.1.200 using FTP with username root
 File Statistics:
```

```
Records awaiting transmission: 1800

Bytes awaiting transmission: 163687

Total records collected: 1800

Total bytes collected: 163687

Total records transmitted: 0

Total bytes transmitted: 0

Total records discarded: 0

Total bytes discarded: 0

Last transfer time required: 0 second(s)

No successful data transfers

Last attempted transfer: Tuesday February 14 15:12:30 EST 2006

File 2 not configured

File 3 not configured

File 4 not configured
```

Manually Gathering and Transferring Bulk Statistics

There may be times where it is necessary to gather and transfer bulk statistics data outside of the configured intervals. The system provides commands that allow you to manually initiate the gathering and transferring of bulk statistics.

These commands are issued from the Exec mode.

To manually initiate the gathering of bulk statistics data outside of the configured sampling interval, enter the following command:

```
bulkstats force gather
```

To manually initiate the transferring of bulk statistics data prior to reaching the of the maximum configured storage limit, enter the following command:

```
bulkstats force transfer
```

Clearing Bulk Statistics Counters and Information

It may be necessary to periodically clear counters pertaining to bulk statistics in order to gather new information or to remove bulk statistics information that has already been collected. The following command can be used to perform either of these functions:

```
clear bulkstats { counters | data }
```

Keyword/Variable	Description
`counters`	Clears the counters maintained by the system's "bulkstats" facility.
`data`	Clears any accumulated data that has not been transferred. This includes any "completed" files that haven't been successfully transferred.

Bulk Statistics Event Log Messages

The stat logging facility provides several events that can be useful for diagnosing errors that could occur with either the creation or writing of a bulk statistic data set to a particular location.

The following table displays information pertaining to these events.

Table 15. Logging Events Pertaining to Bulk Statistics

Event	Event ID	Severity	Additional Information
Local File Open Error	31002	Warning	"Unable to open local file *filename* for storing bulkstats data"
Receiver Open Error	31018	Warning	"Unable to open url *filename* for storing bulkstats data"
Receiver Write Error	31019	Warning	"Unable to write to url *filename* while storing bulkstats data"
Receiver Close Error	31020	Warning	"Unable to close url *filename* while storing bulkstats data"

Chapter 12
Configuring and Viewing System Logs

There are five types of logs that can be configured and viewed on the system:

> ***Important:*** Not all Event Logs can be configured on all products. It is dependent on the hardware platform and licenses in use.

- **Event**: Event logging can be used to determine system status and capture important information pertaining to protocols and tasks in use by the system. This is a global function in that once it is configured, it will be applied to all contexts, sessions, and processes.
- **Trace**: Trace logging can be used to quickly isolate issues that may arise for a particular connected subscriber session. Traces can be taken for a specific call identification (callid) number, IP address, mobile station identification (MSID) number, or username.
- **Active**: Active logs are event logs that are operator configurable on a CLI instance-by-CLI instance basis (i.e. active logs configured by an administrative user in one CLI instance cannot be viewed by an administrative user in a different CLI instance). Each active log can be configured with filter and display properties that are independent of those configured globally for the system. Active logs are displayed in real time as they are generated.
- **Monitor**: Monitor logging records all activity associated with a particular session. This functionality is available in order to comply with law enforcement agency requirements for monitoring capabilities of particular subscribers. Moniors can be performed based on a subscriber's MSID or username.
- **Crash**: Crash logging stores useful information pertaining to system software crashes that may be useful in determining the cause of the crash.

This chapter provides information and instructions for configuring parameters related to the various types of logging and for viewing their content.

Configuring Event Logging Parameters

The system can be configured to generate logs based on user-defined filters. The filters specify the facilities (system tasks or protocols) that the system is to monitor and severity levels at which to trigger the generation of the event log.

Event logs are stored in system memory and can be viewed via the CLI. There are two memory buffers that store event logging information. The first buffer stores the active log information. The second buffer stores inactive logging information. The inactive buffer is used as a temporary repository to allow you to view logs without having data be overwritten. Logs are copied to the inactive buffer only through manual intervention.

Each buffer can store up to 50,000 events. Once these buffers reach their capacity, the oldest information is removed to make room for the newest.

To prevent the loss of log data, the system can be configured to transmit logs to a syslog server over a network interface.

Configuring Event Log Filters

Follow the example below to configure run time event logging parameters for the system:

```
configure
  logging filter runtime facility <facility> level <report_level>
   logging display
   end
```

Notes:

- Configure the logging filter that determines which system facilities should be logged and at what levels.
- Repeat for every facility that you would like to log.
- Option: Configure event ID restrictions by adding the logging disable eventid command. The system provides the ability to restrict the sending of a specific event ID or a range of event IDs to minimize the amount of data logged to that which is most useful. Repeat to disable logging for additional event IDs or event ID ranges.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Syslog Servers

Information generated by the run time event logging filters can be transmitted to a syslog server for permanent storage.

Important: The data transmitted to the syslog server is meant to be used for informational purposes.Therefore, functions such as billing and performance monitoring should not be based on syslogs.

Important: Although the system provides the flexibility to configure syslog servers on a context-by-context basis, it is recommended that all servers be configured in the local context in order to isolate the log traffic from the network traffic.

Use the following example to configure syslog servers:

```
configure
   context local
      logging syslog <ip_address>
      end
```

Notes:

- A number of keyword options/variables are available for the **`logging syslog`** command. Refer to the *Command Line Interface Reference* for more information.
- Repeat as needed to configure additional syslog servers. There is no limit to the number of syslog servers that can be configured.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Trace Logging

Trace logging is useful for quickly resolving issues for specific sessions that are currently active. They are temporary filters that are generated based on a qualifier that is independent of the global event log filter configured using the **logging filter** command. Like event logs, however, the information generated by the logs is stored in the active memory buffer.

All debug level events associated with the selected call are stored.

Important: Trace logs are intrusive to the processing of the session. Therefore, they should be implemented for debug purposes only.

Use the following example to configure trace logs:

```
logging trace { callid <call_id> | ipaddr <ip_address> | msid <ms_id> |
name <username> }
```

Once all of the necessary information has been gathered, the trace log can be deleted by entering the following command:

```
no logging trace { callid <call_id> | ipaddr <ip_address> | msid <ms_id>
| name <username> }
```

Configuring Active Logs

Active logs are event logs that are operator configurable on a CLI instance-by-CLI instance basis (i.e. active logs configured by an administrative user in one CLI instance are not displayed to an administrative user in a different CLI instance). Each active log can be configured with filter and display properties that are independent of those configured globally for the system. Active logs are displayed in real time as they are generated.

Active logs are not written to the active memory buffer by default. To write active logs to the active memory buffer, in the config mode, use the following command:

```
logging runtime buffer store all-events
```

When active logs are written to the active memory buffer, they are available to all users in all CLI instances.

Use the following example to configure active logging:

```
logging filter active facility <facility> level <report_level>

logging active
```

Notes:

- Refer to the **`logging filter`** command in the *Command Line Interface Reference* to view a list of the supported logging facilities.
- A number of keyword options/variables are available for the **`logging active`** command. Refer to the *Command Line Interface Reference* for more information.

Once all of the necessary information has been gathered, the Active log display can be stopped by entering the following command:

```
no logging active
```

Configuring Monitor Logs

Monitor logging records all activity associated with all of a particular subscriber's sessions. This functionality is available in order to comply with law enforcement agency requirements for monitoring capabilities of particular subscribers.

Moniors can be performed based on a subscriber's MSID or username and are only intended to be used for finite periods of time as dictated by the law enforcement agency. Therefore, they should be terminated immediately after the required monitoring period.

This section provides instructions for enabling and disabling monitor logs.

Enabling Monitor Logs

Use the following example to configure monitor log targets:

```
configure
   logging monitor { msid <id> | username <name> | ip_addr | IPv6_addr }
   end
```

Notes:

- Repeat to configure additional monitor log targets.

Disabling Monitor Logs

Use the following example to disable monitor logs:

```
configure
   no logging monitor { msid <id> | username <name> }
   end
```

Viewing Logging Configuration and Statistics

Logging configuration and statistics can be verified by entering the following command from the Exec mode:

```
show logging [ active | verbose ]
```

When no keyword is specified, the global filter configuration is displayed as well as information about any other type of logging that is enabled.

The following table provides information] and descriptions of the statistics that are displayed when the **verbose** keyword is used.

Field	Description
General Logging Statistics	
Total events received	Displays the total number of events generated by the system.
Number of applications receiving events	Displays the number of applications receiving the events.
Logging Source Statistics	
Event sequence ids by process	Displays a list of system processes that have generated events and the reference identification number of the event that was generated.
Msg backlog stat with total cnt	Displays the number of event messages that have been back logged in comparison to the total number of events generated.
LS L2 filter drop rate	Displays the percentage of logging source (LS) layer 2 (L2) event drops.
Abnormal Log Source Statistics	Displays abnormal logging source (LS) statistics, if any.
Runtime Logging Buffer Statistics	
Active buffer	Displays the number of events currently logged in the active memory buffer as well as a date/time timestamp for the oldest and most recent entries in the buffer.
Inactive buffer	Displays the number of events currently logged in the inactive memory buffer.

Viewing Event Logs Using the CLI

Event logs generated by the system can be viewed in one of the following ways:

- **From the syslog server:** If the system is configured to send logs to a syslog server, the logs can be viewed directly on the syslog server.
- **From the system CLI:** Logs stored in the system memory buffers can be viewed directly from the CLI.
- **From the console port:** By default, the system automatically displays events over the console interface to a terminal provided that there is no CLI session active.

This section provides instructions for viewing event logs using the CLI. These instructions assume that you are at the root prompt for the Exec mode.

Step 1 Recommended: Copy the active log memory buffer to the inactive log memory buffer.

When the active log memory buffer is copied to the inactive log memory buffer existing information in the inactive log memory buffer is deleted.

Both active and inactive event log memory buffers can be viewed using the CLI. However, it is preferable to view the inactive log in order to prevent any data from being over-written. The information from the active log buffer can be copied to the inactive log buffer by entering the following command:

```
logs checkpoint
```

Step 2 View the logs by entering the following command:

```
show logs
```

Important: A number of optional keywords/variables are available for the **`show logs`** command. Refer to the *Command Line Interface Reference* for more information.

Configuring and Viewing Software Crash Logging Parameters

In the unlikely even of a software crash, the system stores information that could be useful in determining the reason for the crash. This information can be maintained in system memory or it can be transferred and stored on a network server.

The system supports the generation of the following two types of logs:

- **Crash log:** Crash logs record all possible information pertaining to a software crash. Due to their size, they can not be stored in system memory. Therefore, these logs are only generated if the system is configured with a Universal Resource Locator (URL) pointing to a local device or a network server where the log can be stored.
- **Abridged crash log:** These logs are automatically generated when a software crash occurs and are stored in system memory. The abridged crash log contains a subset of the possible information that could be generated with a crash log. These logs are generated even if a full crash log is generated and can be viewed using the CLI.

Configuring Software Crash Log Destinations

The system can be configured to store software crash log information to any of the following locations:

- **CompactFlash™:** Installed on the SPC/SMC
- **PCMCIA Flash Card:** Installed in either the PCMCIA1 or PCMCIA2 slots on the SPC or in the PCMCIA1 slot on the SMC
- **Network Server:** Any workstation or server on the network that the system can access using the Trivial File Transfer Protocol (TFTP), the File Transfer Protocol (FTP), the Secure File Transfer Protocol (SFTP), or the Hyper-Text Transfer Protocol (HTTP); this is recommended for large network deployments in which multiple systems require the same configuration

Crash logs are stored with unique names as they occur to the specified location. The name format is *crash-card-cpu-time-core*. Where card is the *card* number, *cpu* is the number of the CPU on the card, and *time* is the POSIX timestamp in hexadecimal notation.

Use the following example to configure a software crash log destination:

```
configure
   crash enable [ encrypted ] url <crash_url>
   end
```

Notes:

- Keyword and variable options are available for the crash enable command. Refer to the *Command Line Interface Reference* for more information.

- Repeat to configure additional software crash log destinations. There is no limit to the number of destinations that can be configured.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Viewing Abridged Crash Logs Using the CLI

Abridged crash logs are stored on the CompactFlash installed on the SPC/SMC. They are located in the **/flash/crash/** directory with file names in the *mc-slot-cpu-pid-xxxxxxxx* format. Where *slot* is the card slot in the chassis, *cpu* is the number of the CPU on the card, *pid* is the process ID number, and *xxxxxxxx* is a UNIX date code in hexadecimal notation.

Follow the instructions in this section to view a list of software crashes that have occurred. These instructions assume that you are at the root prompt for the Exec mode.

Step 1 View a listing of any software crashes that may have occurred on the system by entering the following command

Important: The resulting output may not be the same for all platforms:

```
show crash list
```

A sample output is displayed below.

```
== ==== ======= =========== ========= ================

# Time Process Card/CPU/ SW HW_SER_NUM

 PID VERSION Card

== ==== ======= =========== ========= ================

1 2003-Nov-01+11:04:24 kernel 13/00/NA 3.0(3665)
PLX01020114/PLX06020362
```

The following table provides descriptions for the individual columns displayed in the output.

Step 2 ***show crash list Command Output Descriptions***

Column Title	Description
#	Displays an internal reference number for this software crash in the log.
Time	Indicates the date and time that the software crash occurred.
Process	Indicates the software task that experienced the crash.
Card	Indicates the card on which the software task was running.

Column Title	Description
CPU	Indicates the CPU on which the software task was running.
PID	Indicates the process identification (PID) number of the software task that experienced the crash.
SW_Version	Indicates the version of software that experienced the crash.
HW_SER_NUM Card	The hardware serial numbers of the SPC/SMC Card and the Crashed Card.

Step 3 View the abridged crash log by entering the following command:

show crash number <*crash_number*>

crash_number is the number of the crash for which you wish to view the log as displayed by the **show crash list** command.

The information contained in the abridged crash log is useful to help identify and diagnose any internal or external factors causing the software to crash. The following displays a sample of the output.

```
******************** CRASH #30 **********************
Build: 4.0(5800)
Fatal Signal 11: Segmentation fault
 PC: [ 0x484650c] strlen()
 Faulty address: (nil)
 Signal detail: address not mapped to object
 Recent events (oldest first):
 [ 0x38f0498] xtcp_client_timer_tick()
 [ 0x38f0498] xtcp_client_timer_tick()
 [ 0x38f0498] xtcp_client_timer_tick()
 [ 0x38f0498] xtcp_client_timer_tick()
 [ 0x38f0498] xtcp_client_timer_tick()
 [ 0x38f0498] xtcp_client_timer_tick()
 [ 0x38f0498] xtcp_client_timer_tick()
 [ 0x391c630] xtcp_wagg_tick()
 [ 0x391c630] xtcp_wagg_tick()
 [ 0x391c630] xtcp_wagg_tick()
```

```
 [ 0x2c77cb0] snreactor_file_cb()

 [ 0x2c77cb0] snreactor_file_cb()

 [ 0x3932868] sn_epoll_run_events()

 [ 0x3932868] sn_epoll_run_events()

 [ 0x2c77cb0] snreactor_file_cb()

 [ 0x3932868] sn_epoll_run_events()

 Process: card=8 cpu=0 pid=917 argv0=orbs

 Crash time: 2004-Jun-23+12:53:19

 Recent errno: 11 Resource temporarily unavailable

 Registers:

 zr at v0 v1 a0 a1 a2 a3

 00000000 109b20c4 00000000 00000000 00000000 00000000 01010101
80808080

 t0 t1 t2 t3 t4 t5 t6 t7

 00002050 109dbb1c 00000040 00000007 00000000 2abbe9b0 00000000
00000001

 s0 s1 s2 s3 s4 s5 s6 s7

 00000000 7fff6c58 7fff6f38 7fff74a8 7fff74a8 7fff6fc8 7fff7168
00000001

 t8 t9 k0 k1 gp sp s8 ra

 00000000 048464b0 000000f6 00000000 10c1f9f0 7fff6bc0 7fff72d8
048eca80

 Stack: 5192 bytes dumped starting from 0x7fff6850

 [ 0x484650c] strlen() sp=0x7fff6bc0

 [ 0x48eca80] __cxa_bad_typeid() sp=0x7fff6be0

 [0x7fff6f40] <trampoline/gdb/stack>() sp=0x7fff6c10

*******************************************************
```

Saving Log Files

Log files can be saved to a file in a local or remote location specified by a URL. Use the following exec mode command to save log files:

```
save logs { <url> } [ active ] [ inactive ] [ callid <call_id> ] [ event-
verbosity <evt_verboseness>] [ facility <facility> ] [ level <severity_level> ]
[ pdu-data <pdu_format> ] [ pdu-verbosity <pdu_verboseness> ] [ since
<from_date_time> [ until <to_date_time> ] ] [ | { grep <grep_options> | more } ]
```

For detailed information on the **save logs** command, refer to the *Exec Mode Commands* chapter of the *Command Line Interface Reference*.

Event ID Overview

Important: Not all event IDs are used on all platforms. It depends on the platform type and the license(s) running.

Identification numbers (IDs) are used to reference events as they occur when logging is enabled on the system. As described previously in this chapter, logs are collected on a per facility basis. Each facility possesses its own range of event IDs as indicated in the following table.

Table 16. System Event Facilities and ID Ranges

Facility	Event ID Range
A10 Protocol Facility Events	28000-28999
A11 Protocol Facility Events	29000-29999
A11 Manager Facility Events	9000-9999
AAA Client Facility Events	6000-6999
Active Charging Service (ACS) Controller Events	90000-90999
Active Charging Service (ACS) Manager Events	91000-91999
Alarm Controller Facility Events	65000-65999
ASN Gateway Manager Facility Events	100000-100499
ASN Paging/Location-Registry Manager Facility Events	100500-100999
Broadcast/Multicast Service (BCMCS) Facility Events	109000-109999
Call State Control Function (CSCF):	105000-108999:
CSCF CORE module	105000 - 105999
SIPAPP (stack + Proxyapp + B2BUA + Stubs) module	106000 - 106499
Proxy CSCF (P-CSCF) module	106500 - 106549
Serving CSCF (S-CSCF) module	106550 - 106999
CPS	107000 - 107499
CSCF Manager (CSCFMGR)	101000-101999
Card/Slot/Port (CSP) Facility Events	7000-7999
Command Line Interface Facility Events	30000-30999
Content Steering Service Facility Events	77000-77499
Data Path for IPSec Facility Events	54000-54999
Daughter Card Controller Facility Events	62000-62999

Facility	Event ID Range
Daughter Card Manager Facility Events	57000-57999
DHCP Facility Events	53000-53999
Distributed Host Manager Facility Events	83000-83999
Driver Controller Facility Events	39000-39999
eGTP-C Facility Events	141000-141999
eGTP-U Facility Events	142000-142999
eGTP Manager Facility Events	143000-143999
Event Log Facility Events	2000-2999
Foreign Agent Manager Facility Events	33000-33999
GCDR Facility Events	66000-66999
GTP-PRIME Protocol Facility Events	52000-52999
GTPC Protocol Facility Events	47000-47999
GTPC Signaling Demultipler Manager Facility Events	46000-46999
GTPP Storage Server GCDR Facility Events	98000-98099
GTPU Protocol Facility Events	45000-45999
H.248 Protocol Facility Events	42000-42999
High Availability Task Facility Events	3000-3999
Home Agent Manager Facility Events	34000-34999
IMS Authorization Service Library Facility Events	98100-98999
IP Access Control List (ACL) Log Facility Events	21000-21999
IP Address Resolution Protocol (ARP) Facility Events	19000-19999
IP Interface Facility Events	18000-18999
IP Pool Sharing Protocol (IPSP) Facility Events	68000-68999
IP Route Facility Events	20000-20999
IPSec Protocol Facility Events	55000-55999
L2TP Control PDU Protocol Facility Events	50000-50999
L2TP Data PDU Protocol Facility Events	49000-49999
L2TP Demux Facility Events	63000-63999
L2TP Manager Facility Events	48000-48999
Lawful Intercept Log Facility Events	69000-69999
MME App Facility Events	147000-147999
MME Demux Manager Facility Events	154000-154999

Facility	Event ID Range
MME-HSS Facility Events	138000-138999
MME Miscellaneous Facility Events	152600-152999
Mobile Access Gateway Manager Facility Events	137500-137999
Mobile IPv6 Facility Events	129000-129999
Mobile IP Tunneled Data Facility Events	27000-27999
Mobile IP Protocol Facility Events	26000-26999
Multicast Proxy Facility Events	94000-94999
Network Access Signaling Facility Events	153000-153999
Network Storage Events	78000-78999
NPU Control Facility Events	16000-16999
NPU Manager Facility Events	17000-17999
Object Request Broker (ORB) System Facility Events	15000-15999
PDG Facility Events	152010-152999
PDG TCP Demux Manager (pdgdmgr) Facility Events (this is a customer-specific facility)	162400-162999
PDN Gateway Facility Events	139000-139999
Point-To-Point Protocol Facility Events	25000-25999
RADIUS Accounting Protocol Facility Events	24000-24999
RADIUS Authentication Protocol Facility Events	23000-23999
Recovery Control Task (RCT) Facility Events	13000-13999
Redirector Task (RDT) Facility Events	67000-67999
Resource Manager (RM) Facility Events	14000-14999
Robust Header Compression Protocol (ROHC) Facility Events	103000-103999
RSVP Protocol Facility Events	93000-93999
Service Redundancy Protocol (SRP) Facility Events	84000-84999
Serving Gateway Facility Events	140000-140999
Session Controller Facility Events	8000-8999
Session Initiation Protocol (SIP) Events	59000-59999
Session Manager Facility Events	10000-12999
Shared Configuration Task (SCT) Facility Events	32000-32099
Session Initiation Protocol (SIP) Call Distributor Facility Events	86000-86999
Simple Network Management Protocol (SNMP) Protocol Facility Events	22000-22999
SSL Facility Events (this is a customer-specific facility)	156200-157199

Facility	Event ID Range
Static Rating Database Facility Events	102000-102999
Statistics Facility Events	31000-31999
Switch Fabric Task (SFT) Facility Events	58000-58999
System Facility Events	1000-1999
System Initiation Task (SIT) Main Facility Events	4000-4999
TACACS+ Protocol Facility Events	37000-37999
Threshold Facility Events	61000-61999
TTG Facility Events	130000-130999
UDR Facility Events	79000-79999
User-Data Facility Events	51000-51999
User L3 Tunnel Facility Events	75000-75999
Virtual Private Network Facility Events	5000-5999
WiMAX R6 Protocol (Signaling) Facility	104000-104899

Event Severities

The system provides the flexibility to configure the level of information that is displayed when logging is enabled. The following levels are supported:

- **critical:** Logs only those events indicating a serious error has occurred that is causing the system tor a system component to cease functioning. This is the highest severity level.
- **error:** Logs events that indicate an error has occurred that is causing the system or a system component to operate in a degraded state. This level also logs events with a higher severity level.
- **warning:** Logs events that may indicate a potential problem. This level also logs events with a higher severity level.
- **unusual:** Logs events that are very unusual and may need to be investigated. This level also logs events with a higher severity level.
- **info:** Logs informational events and events with a higher severity level.
- **trace:** Logs events useful for tracing and events with a higher severity level.
- **debug:** Logs all events regardless of the severity.

Each of the above levels correspond to the “severity” level of the event ID. Therefore, only those event IDs with a “severity” level equal to the logging level are displayed.

Understanding Event ID Information in Logged Output

This section explains the event information that is displayed when logging is enabled.

The following displays a sample output for an event that was logged.

```
2006-Jun-23+12:18:41.993 [cli 30005 info] [8/0/609 <cli:8000609>
_commands_cli.c:1290] [software internal system] CLI session ended for
Security Administrator admin on device /dev/pts/2
```

The following table describes the elements of contained in the sample output.

Table 17. Event Element Descriptions

Element	Description
2006-Jun-23+12:18:41.993	Date/Timestamp indicating when the event was generated
[cli 30005 info]	Information about the event including: • The facility the event belongs to • The event ID • The event's severity level In this example, the event belongs to the CLI facility, has an ID of 3005, and a severity level of "info".
[8/0/609 <cli:8000609> _commands_cli.c:1290]	Information about the specific CLI instance.
[software internal system]	Indicates that the event was generated because of system operation.
CLI session ended for Security Administrator admin on device /dev/pts/2	The event's details. Event details may, or may not include variables that are specific to the occurrence of the event.

Chapter 13
Troubleshooting the System

This chapter provides information and instructions for using the system command line interface (CLI) for troubleshooting any issues that may arise during system operation.

The following topics are included:

Detecting Faulty Hardware Using Component LEDs

Upon applying power to the chassis, power is provided to the upper and lower fan trays, and every application and line card that is installed.

Each PFU, application, and line card installed in the system has light emitting diodes (LEDs) that indicate it's status. This section provides information and instructions pertaining to using LEDs to verify for verifying that all of the installed components are functioning properly.

Important: As the system progresses through its boot process, some cards may have no immediate LED activity. It is recommended that several minutes elapse prior to checking the LEDs on the various cards to verify the installation.

Using the CLI to View Component LEDs

The status of application and line card LEDs can be viewed through the CLI by entering the following command:

show leds all

The following displays a sample of this command's output.

```
Slot 01: Run/Fail: Green | Active: Off | Standby: Green
Slot 08: Run/Fail: Green | Active: Green | Standby: Off
 Status: Green | Service: Off |
Slot 09: Run/Fail: Green | Active: Off | Standby: Green
 Status: Green | Service: Off |
Slot 12: Run/Fail: Green | Active: Green | Standby: Off
Slot 14: Run/Fail: Green | Active: Green | Standby: Off
Slot 17: Run/Fail: Green | Active: Green | Standby: Off
Slot 24: Run/Fail: Green | Active: Green | Standby: Off
Slot 25: Run/Fail: Green | Active: Off | Standby: Green
Slot 30: Run/Fail: Green | Active: Green | Standby: Off
Slot 33: Run/Fail: Green | Active: Off | Standby: Off
Slot 40: Run/Fail: Green | Active: Green | Standby: Off
```

The status of the chassis' two Power Filter Units (PFUs) can be viewed by entering the following command:

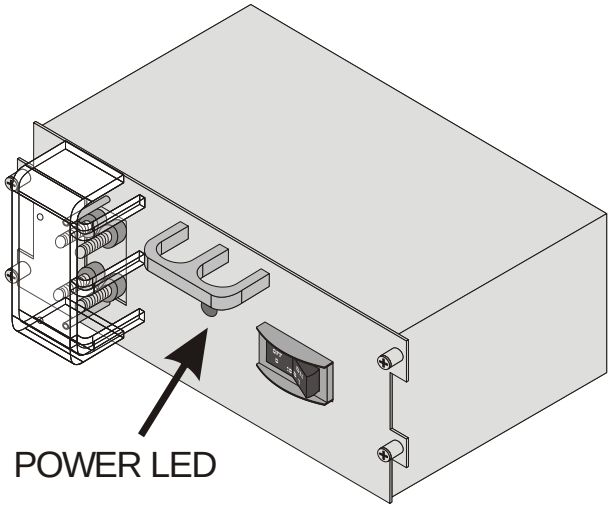
```
show power chassis
```

Checking the LED on the PFU(s)

Each PFU has a single status LED labeled POWER.

This LED should be green for normal operating conditions.

Figure 12. PFU LED Location



The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information below to diagnose the problem.

Table 18. PFU POWER LED States

Color	Description	Troubleshooting
Green	PFU powered with no errors detected	None needed.

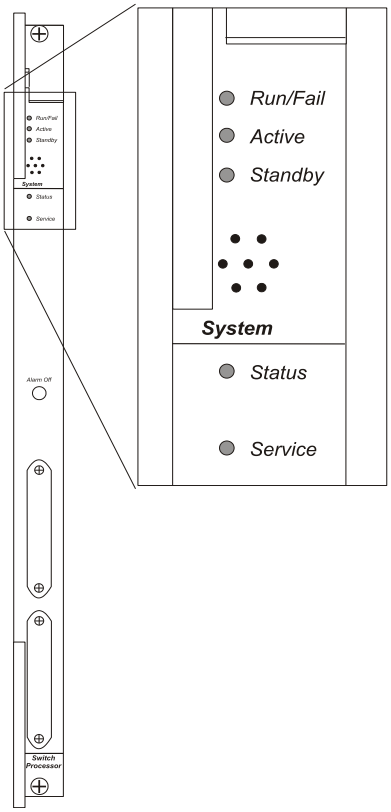
Color	Description	Troubleshooting
None	PFU is not receiving power	• Verify that the power switch is in the ON position. • Verify that the RTN and -VDC lugs are attached properly according to the instructions provided in this document. • Verify that the ground lug is attached properly.Verify that the power source is on and is supplying the correct voltage and sufficient current. • Check the cables from the power source to the rack for continuity. • If a fuse panel is installed between the power distribution frame (PDF) and the chassis, verify that the fuses are intact. • If a fuse panel is installed between the PDF and the chassis, check the cables from the fuse panel to the chassis for continuity. • If all of the above suggestions have been verified, then it is likely that the PFU is not functional. Please contact your service representative.

Checking the LEDs on the SPC(s)

Each SPC is equipped with the following LEDs :

- RUN/FAIL
- Active
- Standby
- Status
- Service

Figure 13. SPC LED Locations



The possible states for all SPC LEDs are described in the sections that follow.

SPC RUN/FAIL LED States

The SPC's *RUN/FAIL* LED indicates the overall status of the card. This LED should be green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 19. SPC RUN/FAIL LED States

Color	Description	Troubleshooting
Green	Card powered with no errors detected	None needed.
Blinking Green	Card is initializing and/or loading software	This is normal operation during boot-up.
Red	Card powered with error(s) detected	Errors were detected during the POSTs. It is likely that the errors were logged to the system's command line interface during boot.

Color	Description	Troubleshooting
None	Card is not receiving power	• Verify that the *POWER* LEDs on the PFUs are green. If they are not, refer to the *Checking the LED on the PFU(s)* section in this chapter for troubleshooting information. • Verify that the power source is supplying ample voltage and current to the chassis. • Verify that the card is properly installed per the instructions in the *Hardware Installation and Administration Guide*. • If all of the above suggestions have been verified, it is possible that the SPC is not functional. Please contact your service representative.

SPC Active LED States

The *Active* LED on the SPC indicates that the software is loaded on the card and it is ready for operation. For the SPC installed in chassis slot 8, this LED should be green for normal operation. For the SPC installed in slot 9, this LED should be off for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 20. SPC Active LED States

Color	Description	Troubleshooting
Green	Card is active	None needed for the SPC in slot 8. If green for the SPC in slot 9, verify that the SPC in slot 8 is installed properly according to the instructions in this document.
Blinking Green	Tasks or processes being migrated from the active SPC to the redundant/secondary SPC	Verify that the *Standby* LED on the redundant SPC is also blinking green. If so, there is an issue with the active SPC. Refer to the *Monitoring the System* chapter of this reference for information on determining the status of the SPC and/or for information determining the status of system software processes and functionality.
None	Card is not receiving power**OR**Card in Standby Mode	• Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the *SPC RUN/FAIL LED States* section of this chapter for troubleshooting information. • Check the state of the *Standby* LED. If it is green, the card is in standby mode. This is normal for the SPC in slot 9 since it is automatically placed into standby mode at boot up.

SPC Standby LED States

The *Standby* LED on the SPC indicates that software is loaded on the card, but it is serving as a redundant component. For the SPC installed in slot 9, this LED should be green for normal operation. For the SPC installed in slot 8, this LED should be off for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 21. SPC Standby LED States

Color	Description	Troubleshooting
Green	Card is in redundant mode	None needed for the SPC in slot 9. If green for the SPC in slot 8, then verify it is installed properly according to the instructions in this document.
Blinking Green	Tasks or processes being migrated from the active SPC to the redundant/secondary SPC	Verify that the *Active* LED on the redundant SPC is also blinking green. If so, there is an issue with the active SPC. Refer to the *Monitoring the System* chapter of this reference for information on determining the status of the SPC and system software processes and functionality.
None	Card is not receiving power**OR**Card in Active Mode	• Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the *SPC RUN/FAIL LED States* section of this chapter for troubleshooting information on. • Check the state of the *Active* LED. If it is green, the card is in active mode. This is normal for the SPC in slot 8 since it is automatically made active at boot up.

SPC Status LED States

The *Status* LEDs on the SPC indicate the status of system level hardware such as installed cards, fans, and PFUs. This LED is green during normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information also provided to diagnose the problem.

Table 22. SPC Status LED States

Color	Description	Troubleshooting
Green	No system errors detected	None needed.

Color	Description	Troubleshooting
Red	Failures Detected	• Check the *RUN/FAIL* LEDs for all installed application cards, and line cards. If any are red or off, refer to the troubleshooting information in this chapter pertaining to that device. • Refer to the *Monitoring the System* chapter of this reference for information on determining the status of system hardware components.
None	Card is not receiving power	Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the *SPC RUN/FAIL LED States* section of this chapter for troubleshooting information.

SPC Service LED States

The *Service* LEDs on the SPCs indicate that the system requires maintenance or service (e.g. the system could not locate a a valid software image at boot-up, or a high temperature condition exists).

This LED is off during normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 23. SPC Service LED States

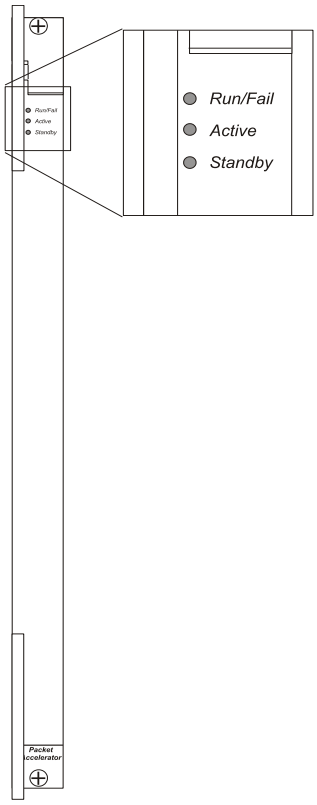
Color	Description	Troubleshooting
Yellow	System requires maintenance (fan filter, temperature warning, PFU outage, etc.)	Refer to the *Monitoring the System* chapter of this reference for information on determining the status of system hardware components and system software processes and functionality.
None	Card is not receiving power	No maintenance needed.

Checking the LEDs on the PAC(s)

Each PAC is equipped with status LEDs as listed below:

- RUN/FAIL
- Active
- Standby
- Status
- Service

Figure 14. PAC LED Locations



The possible states for all of the PAC's LEDs are described in the sections that follow.

PAC RUN/FAIL LED States

The PAC's *RUN/FAIL* LED indicates the overall status of the card. This LED should be green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 24. PAC RUN/FAIL LED States

Color	Description	Troubleshooting
Green	Card powered with no errors detected	None needed.
Blinking Green	Card is initializing and/or loading software	None needed.

Color	Description	Troubleshooting
Red	Card powered with error(s) detected	Errors were detected during the POSTs. It is likely that the errors were logged to the system's command line interface during the boot process. Refer to the *Monitoring the System* chapter of this reference for information on determining the status of system hardware components.
None	Card is not receiving power	• Verify that the *POWER* LEDs on the PFUs are green. If they are not, refer to the *Checking the LED on the PFU(s)* section in this chapter for troubleshooting information. • Verify that the power source is supplying ample voltage and current to the chassis. • Verify that the card is properly installed per the instructions in the *Hardware Installation and Administration Guide*. • If all of the above suggestions have been verified, it is possible that the PAC is not functional. Please contact your service representative.

PAC Active LED States

The *Active* LED on the PAC indicates that the software is loaded on the card and that the card is ready for operation.When the system first boots up, all installed PACs are booted into standby mode. The system must then be configured as to which PACs should serve as redundant components (i.e., remain in standby mode) and which should function as active components.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 25. PAC Active LED States

Color	Description	Troubleshooting
Green	Card is active	The first time power is applied to the system, all of the PACs should be booted into the standby mode. Therefore, this LED should be off.
Blinking Green	Tasks or processes being migrated from an active PAC to a redundant/secondary PAC	Verify that the *Standby* LED on a redundant PAC is also blinking green. If so, there is an issue with the PAC that was active and is transferring its processes. Refer to the *Monitoring the System* chapter of this reference for information on determining the status of the PAC and system software processes and functionality.

Color	Description	Troubleshooting
None	Card is not receiving powerORCard in Standby Mode	• Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the *PAC RUN/FAIL LED States* section of this chapter for troubleshooting information. • Check the state of the *Standby* LED. If it is green, the card is in standby mode. This is normal operation for the initial power-up. If needed, refer to the *Configuring* PAC/*PSC and Line Card Availability* section of the *Configuring System Settings* chapter of this reference for information on making the card active.

PAC Standby LED States

The *Standby* LED on the PAC indicates that software is loaded on the card, but the card is serving as a redundant component. When the system first boots up, all installed PACs are booted into standby mode. The system must then be configured as to which PACs should serve as redundant components (i.e. remain in standby mode) and which should function as active components.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 26. PAC Standby LED States

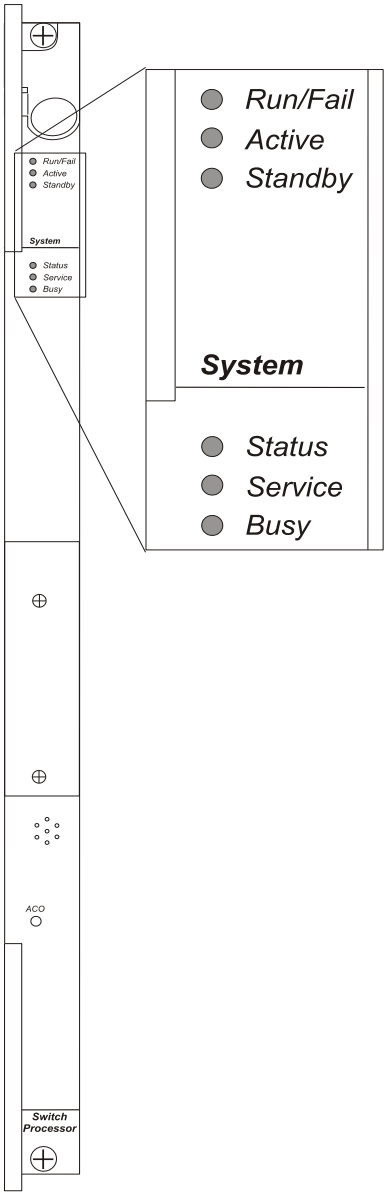
Color	Description	Troubleshooting
Green	Card is in redundant mode	The first time power is applied to the system, all of the PACs should be booted into the standby mode. Therefore, this is normal operation.
Blinking Green	Tasks or processes being migrated from the active PAC to the redundant/secondary PAC	Verify that the *Active* LED on the redundant PAC is also blinking green. If so, there is an issue with the active PAC and the system is transferring its processes. Refer to the *Monitoring the System* chapter of this reference for information on determining the status of the PAC and system software processes and functionality.
None	Card is not receiving powerORCard in Active Mode	• Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the *PAC RUN/FAIL LED States* section of this chapter for information on troubleshooting. • Check the state of the *Active* LED. If it is green, the card is in active mode.

Checking the LEDs on the SMC(s)

Each SMC is equipped with the following LEDs as shown in the following figure:

- RUN/FAIL
- Active
- Standby
- Status
- Service
- Busy

Figure 15. SMC LEDs



The possible states for all SMC LEDs are described in the sections that follow.

SMC RUN/FAIL LED States

The SMC *RUN/FAIL* LED indicates the overall status of the card. This LED should be green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 27. SMC RUN/FAIL LED States

Color	Description	Troubleshooting
Green	Card powered with no errors detected	None needed.
Blinking Green	Card is initializing and/or loading software	This is normal operation during boot-up.
Red	Card powered with error(s) detected	Errors were detected during the POSTs. It is likely that the errors were logged to the system's command line interface during boot. Refer to one or more of the following to help analyze this problem: • The *Monitoring Hardware Status* chapter in this reference for show commands; the outputs of which will assist in further determining the problem. • The *Configuring and Viewing System Logs* chapter in this reference for information on configuring specific types of logging information and how to view logs. • *SNMP MIB Reference* for information on associated status and alarm conditions.
None	Card is not receiving power	• Verify that the *POWER* LEDs on the PFUs are green. If they are not, refer to the *Checking the LEDs on the SPC(s)* section in this chapter for troubleshooting information. • Verify that the power source is supplying ample voltage and current to the chassis. • Verify that the card is properly installed per the instructions in the *Hardware Installation and Administration Guide*. • If all of the above suggestions have been verified, it is possible that the SMC is not functional. Please contact your service representative.

SMC Active LED States

The *Active* LED on the SMC indicates that the software is loaded on the card and it is ready for operation. For the SMC installed in chassis slot 8, this LED should be green for normal operation. For the SMC installed in slot 9, this LED should be off for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 28. SMC Active LED States

Color	Description	Troubleshooting
Green	Card is active	None needed for the SMC in slot 8. If green for the SMC in slot 9, verify that the SMC in slot 8 is installed properly according to the instructions in this document.

Color	Description	Troubleshooting
Blinking Green	Tasks or processes being migrated from the active SMC to the redundant/secondary SMC	Verify that the *Standby* LED on the redundant SMC is also blinking green. If so, there is an issue with the active SMC. Refer to one or more of the following to help analyze this problem: • The *Monitoring Hardware Status* chapter in this reference for show commands; the outputs of which will assist in further determining the problem. • The *Configuring and Viewing System Logs* chapter in this reference for information on configuring specific types of logging information and how to view logs. • *SNMP MIB Reference* for information on associated status and alarm conditions.
None	Card is not receiving power**OR**Card in Standby Mode	• Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, please refer to the *SMC RUN/FAIL LED States* section of this chapter for troubleshooting information. • Check the state of the *Standby* LED. If it is green, the card is in standby mode. If needed, refer to the *Configuring* PAC/*PSC and Line Card Availability* section of the Configuring System Settings chapter in this reference for information on making the card active.

SMC Standby LED States

The *Standby* LED on the SMC indicates that software is loaded on the card, but it is serving as a redundant component. For the SMC installed in slot 9, this LED should be green for normal operation. For the SMC installed in slot 8, this LED should be off for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 29. SMC Standby LED States

Color	Description	Troubleshooting
Green	Card is in redundant mode	None needed for the SMC in slot 9. If green for the SMC in slot 8, then verify it is installed properly according to the instructions in this document.

Color	Description	Troubleshooting
Blinking Green	Tasks or processes being migrated from the active SMC to the redundant/secondary SMC	Verify that the *Active* LED on the redundant SMC is also blinking green. If so, there is an issue with the active SMC. Refer to one or more of the following to help analyze this problem: • The *Monitoring Hardware Status* chapter in this reference for show commands; the outputs of which will assist in further determining the problem. • The *Configuring and Viewing System Logs* chapter in this reference for information on configuring specific types of logging information and how to view logs. • *SNMP MIB Reference* for information on associated status and alarm conditions.
None	Card is not receiving power**OR**Card in Active Mode	• Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, please refer to the *SMC RUN/FAIL LED States* section of this chapter for troubleshooting information on. • Check the state of the *Active* LED. If it is green, the card is in active mode. If needed, refer to the *Manually Initiating an* SPC/*SMC Switchover* section in this chapter for information on configuring the card to serve as a redundant component.

SMC Status LED States

The *Status* LEDs on the SMC indicate the status of system level hardware such as installed cards, fans, and PFUs. This LED is green during normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information also provided to diagnose the problem.

Table 30. SMC Status LED States

Color	Description	Troubleshooting
Green	No system errors detected	None needed.

Color	Description	Troubleshooting
Red	Failures Detected	• Check the *RUN/FAIL* LEDs for all installed application cards, and line cards. If any are red or off, refer to the troubleshooting information in this chapter pertaining to that device. • Refer to one or more of the following to help analyze this problem: • The *Monitoring Hardware Status* chapter in this reference for show commands; the outputs of which will assist in further determining the problem. • The *Configuring and Viewing System Logs* chapter in this reference for information on configuring specific types of logging information and how to view logs. • *SNMP MIB Reference* for information on associated status and alarm conditions.
None	Card is not receiving power	Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the *SMC RUN/FAIL LED States* section of this chapter for troubleshooting information.

SMC Service LED States

The *Service* LEDs on the SMCs indicate that the system requires maintenance or service (e.g. the system could not locate a a valid software image at boot-up, or a high temperature condition exists).

This LED is off during normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 31. SMC Service LED States

Color	Description	Troubleshooting
Yellow	System requires maintenance (fan filter, temperature warning, PFU outage etc.)	Refer to one or more of the following to help analyze this problem: • The *Monitoring Hardware Status* chapter in this reference for show commands; the outputs of which will assist in further determining the problem. • The *Configuring and Viewing System Logs* chapter in this reference for information on configuring specific types of logging information and how to view logs. • *SNMP MIB Reference* for information on associated status and alarm conditions.
None	Card is not receiving power	No maintenance needed.

SMC Busy LED States

The *Busy* LEDs on the SMCs indicate that there is activity on one of their memory devices. Activity is displayed for the following memory devices:

- CompactFlash module
- PCMCIA device
- Nand Flash (used to store SMC firmware).
- Hard Drive

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 32. SMC Busy LED States

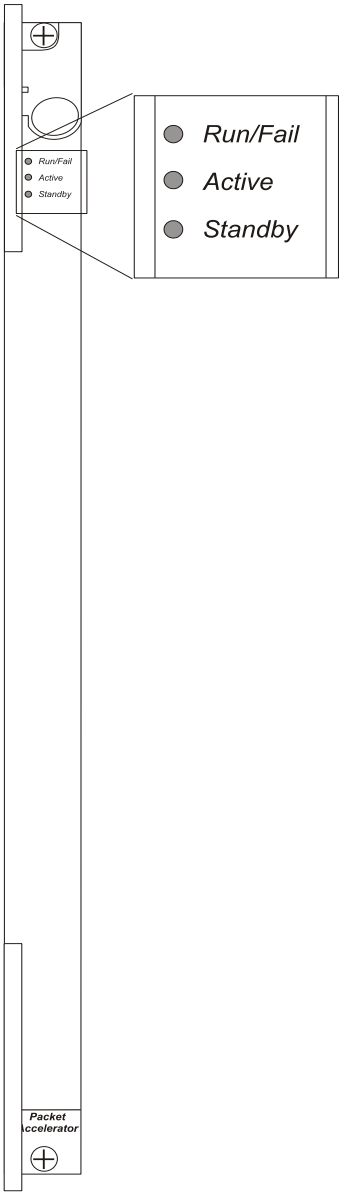
Color	Description	Troubleshooting
Green/ Blinking Green	Data is being read from/written to one of the memory devices.	No maintenance needed. ***Important:*** If removing the SMC from the chassis, it is recommended that you wait until this LED is off to ensure the integrity of all data being transferred to or from the memory device.
None	The memory devices are not in use.	No maintenance needed.

Checking the LEDs on the PSC(s)

Each PSC is equipped with status LEDs as listed below:

- RUN/FAIL
- Active
- Standby
- Status
- Service

Figure 16. PSC LEDs



The possible states for all PSC LEDs are described below:

PSC RUN/FAIL LED States

The PSC *RUN/FAIL* LED indicates the overall status of the card. This LED should be green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 33. PSC RUN/FAIL LED States

Color	Description	Troubleshooting
Green	Card powered with no errors detected	None needed.
Blinking Green	Card is initializing and/or loading software	None needed.
Red	Card powered with error(s) detected	Errors were detected during the POSTs. It is likely that the errors were logged to the system's command line interface during the boot process. Refer to one or more of the following to help analyze this problem: • The *Monitoring Hardware Status* chapter in this reference for show commands; the outputs of which will assist in further determining the problem. • The *Configuring and Viewing System Logs* chapter in this reference for information on configuring specific types of logging information and how to view logs. • *SNMP MIB Reference* for information on associated status and alarm conditions.
None	Card is not receiving power	• Verify that the *POWER* LEDs on the PFUs are green. If they are not, refer to the *Checking the LED on the PFU(s)* section in this chapter for troubleshooting information. • Verify that the power source is supplying ample voltage and current to the chassis. • Verify that the card is properly installed per the instructions in the *Hardware Installation and Administration Guide*. • If all of the above suggestions have been verified, it is possible that the PSC is not functional. Please contact your service representative.

PSC Active LED States

The *Active* LED on the PSC indicates that the software is loaded on the card and that the card is ready for operation.When the system first boots up, all installed PSCs are booted into standby mode. The system must then be configured as to which PSCs should serve as redundant components (i.e. remain in standby mode) and which should function as active components.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 34. PSC Active LED States

Color	Description	Troubleshooting

Color	Description	Troubleshooting
Green	Card is active	The first time power is applied to the system, all of the PSCs should be booted into the standby mode. Therefore, this LED should be off.
Blinking Green	Tasks or processes being migrated from an active PSC to a redundant/secondary PSC	Verify that the *Standby* LED on a redundant PSC is also blinking green. If so, there is an issue with the PSC that was active and is transferring its processes. Refer to the *Monitoring the System* chapter of this reference for information on determining the status of the PSC and system software processes and functionality.
None	Card is not receiving powerORCard in Standby Mode	• Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, please refer to the *PSC RUN/FAIL LED States* section of this chapter for troubleshooting information. • Check the state of the *Standby* LED. If it is green, the card is in standby mode. This is normal operation for the initial power-up. If needed, refer to the *Configuring* PAC/*PSC and Line Card Availability* section of the *Configuring System Settings* chapter in this reference for information on making the card active.

PSC Standby LED States

The *Standby* LED on the PSC indicates that software is loaded on the card, but the card is serving as a redundant component. When the system first boots up, all installed PSCs are booted into standby mode. The system must then be configured as to which PSCs should serve as redundant components (i.e. remain in standby mode) and which should function as active components.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 35. PSC Standby LED States

Color	Description	Troubleshooting
Green	Card is in redundant mode	The first time power is applied to the system, all of the PSCs should be booted into the standby mode. Therefore, this is normal operation.

Color	Description	Troubleshooting
Blinking Green	Tasks or processes being migrated from the active SMC to the redundant/secondary SMC	Verify that the *Active* LED on the redundant PSC is also blinking green. If so, there is an issue with the active PSC and the system is transferring its processes. Refer to one or more of the following to help analyze this problem: • The *Monitoring Hardware Status* chapter in this reference for show commands; the outputs of which will assist in further determining the problem. • The *Configuring and Viewing System Logs* chapter in this reference for information on configuring specific types of logging information and how to view logs. • *SNMP MIB Reference* for information on associated status and alarm conditions.
None	Card is not receiving powerORCard in Active Mode	• Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, please refer to the *PSC RUN/FAIL LED States* section of this chapter for information on troubleshooting. • Check the state of the *Active* LED. If it is green, the card is in active mode. If needed, refer to the *Manually Initiating a* PAC/*PSC Migration* section in this chapter for information on configuring the card to serve as a redundant component.

Checking the LEDs on the SPIO(s)

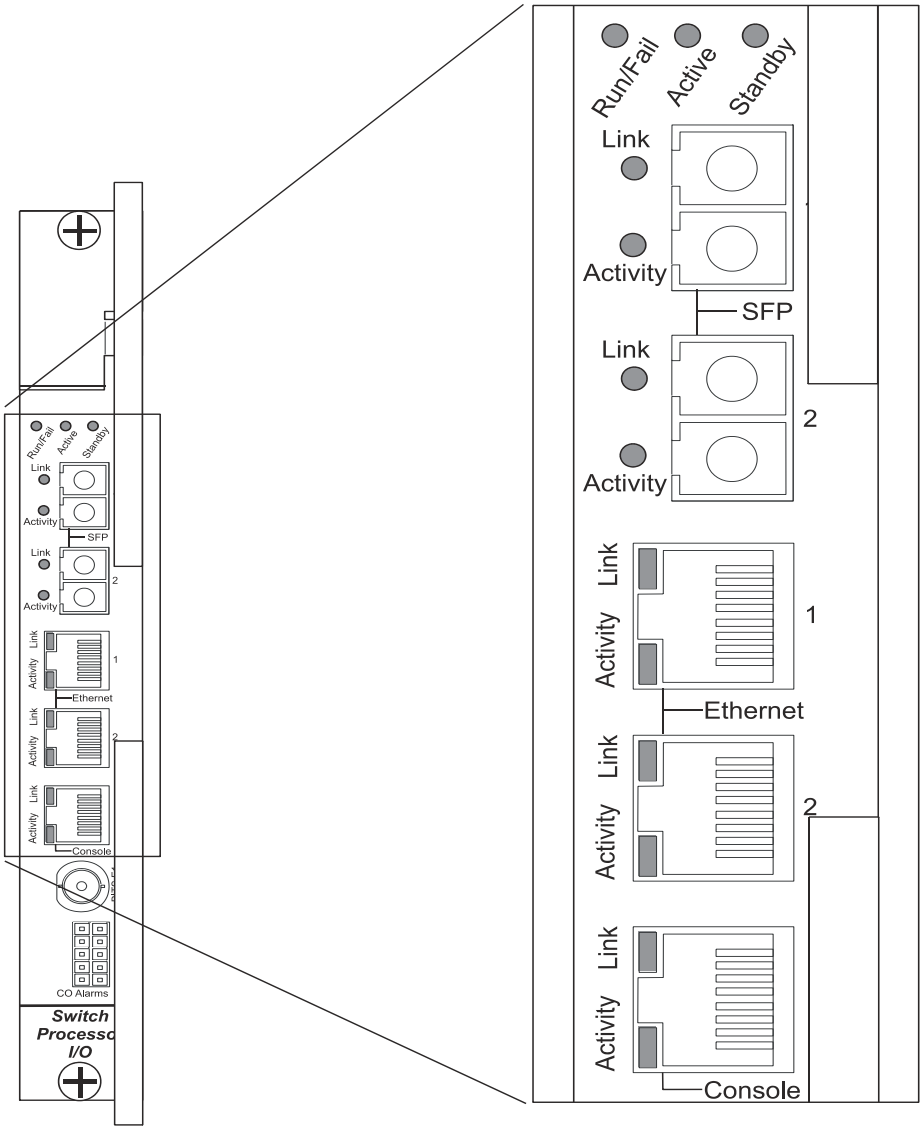
Each SPIO is equipped with status LEDs as listed below:

- RUN/FAIL
- Active
- Standby

In addition to the LEDs listed above, each interface to the management network (both RJ-45 and SFP) are equipped with the following LEDs:

- Link
- Activity

Figure 17. SPIO LED Locations



The possible states for all of the SPIO's LEDs are described in the sections that follow.

SPIO RUN/FAIL LED States

The SPIO's *RUN/FAIL* LED indicates the overall status of the card. This LED should be green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 36. SPIO RUN/FAIL LED States

Color	Description	Troubleshooting
Green	Card powered with no errors detected	None needed.
Red	Card powered with error(s) detected	Errors were detected during the POSTs. It is likely that the errors were logged to the system's command line interface during the boot process. Refer to the *Monitoring the System* chapter of this reference for information on determining the status of system hardware components.
None	Card is not receiving power	• Verify that the *POWER* LEDs on the PFUs are green. If they are not, refer to the *Checking the LED on the PFU(s)* section in this chapter for troubleshooting information. • Verify that the power source is supplying ample voltage and current to the chassis. • Verify that the card is properly installed per the instructions in the *Hardware Installation and Administration Guide*. • If all of the above suggestions have been verified, it is possible that the SPIO is not functional. Please contact your service representative.

SPIO Active LED States

The *Active* LED on the SPIO indicates that the software is loaded on the card and that the card is ready for operation. For the SPIO installed in chassis slot 24, this LED should be green for normal operation. For the SPIO installed in slot 25, this LED should be off for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 37. SPIO Active LED States

Color	Description	Troubleshooting
Green	Card is active	None needed for SPIO in slot 24. If green for SPIO in slot 25, then verify that SPIO in slot 24 is installed properly according to the instructions in this document.
None	Card is not receiving power **OR** Card in Standby Mode	• Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the *SPIO RUN/FAIL LED States* section of this chapter for troubleshooting information. • Check the state of the *Standby* LED. If it is green, the card is in standby mode. This is normal for the SPIO in slot 25 since the chassis automatically places the card into standby mode at boot up.

SPIO Standby LED States

The *Standby* LED on the SPIO indicates that software is loaded on the card, but it is serving as a redundant component. For the SPIO installed in slot 25, this LED should be green for normal operation. For the SPIO installed in slot 24, this LED should be off for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 38. SPIO Standby LED States

Color	Description	Troubleshooting
Green	Card is in redundant mode	None needed for SPIO in slot 25. If green for SPIO in slot 24, check the status of the SPC/SMC installed in slot 8. If the SPC/SMC in slot 8 is in standby mode, it is possible that there is a problem. Refer to one or more of the following to help analyze this problem: • The *Monitoring Hardware Status* chapter in this reference for show commands; the outputs of which will assist in further determining the problem. • The *Configuring and Viewing System Logs* chapter in this reference for information on configuring specific types of logging information and how to view logs. • *SNMP MIB Reference* for information on associated status and alarm conditions.
None	Card is not receiving power **OR** Card in Active Mode	• Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the *SPIO RUN/FAIL LED States* section of this chapter for information on troubleshooting. • Check the state of the *Active* LED. If it is green, the card is in active mode. This is normal for the SPIO in slot 24 since the chassis automatically make the card active at boot up.

SPIO Interface Link LED States

The *Link* LED, associated with a particular SPIO interface indicates the status of the network link. This LED should be green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 39. SPIO Interface Link LED States

Color	Description	Troubleshooting

Color	Description	Troubleshooting
Green	Link is up	None needed. **Important:** This LED will not indicate the presence of a network link until the interface parameters are set during the software configuration process.
None	No power to card **OR** Link is down	• Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power. If it is off, refer to the *SPIO RUN/FAIL LED States* section of this chapter for troubleshooting information. • Verify that the interface is cabled properly. • Verify that the device on which the interface is located is cabled and powered properly.

SPIO Interface Activity LED States

The *Activity* LED associated with a particular SPIO interface indicates the presence of traffic on the network link. This LED should be green when data is being transmitted or received over the interface.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 40. SPIO Interface Activity LED States

Color	Description	Troubleshooting
Flashing Green	Traffic is present on the link	None needed.
None	No traffic is present on the link	None needed if there is no activity on the link. Prior to configuration, this is normal operation.

Checking the LEDs on the Ethernet 10/100 and Ethernet 1000/Quad Gig-E Line Card(s) (QGLC)

Each Ethernet 10/100, Ethernet 1000 Line Card and QGLC is equipped with status LEDs as listed below:

- RUN/FAIL

- Active
- Standby

In addition to the LEDs listed above, each network interface is equipped with the following LEDs:

- Link
- Activity

Figure 18. Ethernet 10/100 and GigE LEDs

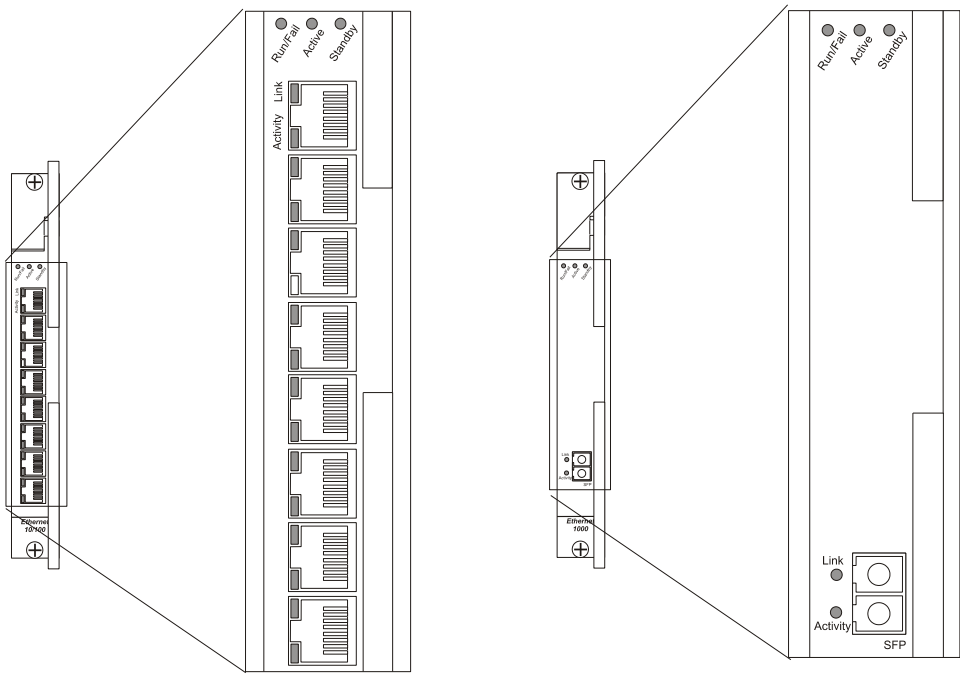
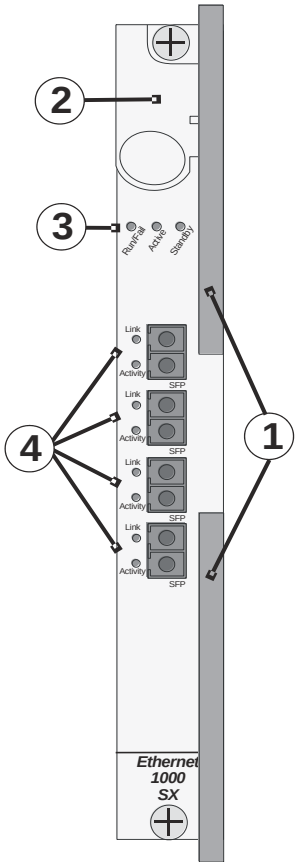


Figure 19. QGLC LEDs



The possible states for all LEDs on the Ethernet 10/100 and Ethernet 1000/QGLC cards are as follows:

Ethernet 10/100 and Ethernet 1000/QGLC RUN/FAIL LED States

The *RUN/FAIL* LEDs on the Ethernet 10/100 and Ethernet 1000/QGLC Line Cards indicate the overall status of the cards. These LEDs should be green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 41. Ethernet 10/100 and Ethernet 1000 RUN/FAIL LED States

Color	Description	Troubleshooting
Green	Card powered with no errors detected	None needed.

Color	Description	Troubleshooting
Red	Card powered with error(s) detected	Errors were detected during the POSTs. It is likely that the errors were logged to the system's command line interface during the boot process. Refer to the *Monitoring the System* chapter of this reference for information on determining the status of system hardware components.
None	Card is not receiving power	• Verify that the *POWER* LEDs on the PFUs are green. If they are not, refer to the *Checking the LED on the PFU(s)* section in this chapter for troubleshooting information. • Verify that the power source is supplying ample voltage and current to the chassis. • Verify that the card is properly installed per the instructions in the *Hardware Installation and Administration Guide*. • If all of the above suggestions have been verified, it is possible that the line card is not functional. Please contact your service representative.

Ethernet 10/100 and Ethernet 1000/QGLC Active LED States

The *Active* LEDs on the Ethernet 10/100 and Ethernet 1000/QGLC Line Cards indicate that the operating software is loaded on the card and that the card is ready for operation.

Important: Quad Gigabit Ethernet line cards (QGLC) only work in an ASR 5000 behind a PSC.

The line cards installed will remain in a ready mode until their corresponding PAC/PSC is made active via configuration. While in ready mode the Active LED should be off. After the PAC/PSC is made active, the line card installed in the upper-rear chassis slot behind the PAC/PSC will also be made active. The line card installed in the lower-rear chassis slot behind the PAC/PSC will enter the standby mode.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 42. Ethernet 10/100 and Ethernet 1000/QGLC Active LED States

Color	Description	Troubleshooting
Green	Card is active	None needed for line cards installed in slots 17 through 23 and 26 through 32 after configuration. If green for line cards in slots 33 through 39 and 42 through 48, verify that the corresponding line card installed in the upper-rear chassis slot is installed properly according to the instructions in this document. For example, if this LED is green for a line card in slot 33, verify that the line card in slot 17 is installed properly.

Color	Description	Troubleshooting
None	Card in Ready ModeORCard is not receiving power**OR**Card in Standby Mode	• This is normal prior to configuration. Neither the Active or the *Standby* LED on the card is on. • Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the *Ethernet 10/100 and Ethernet 1000/QGLC RUN/FAIL LED States* section of this chapter for troubleshooting information. • Check the state of the *Standby* LED. If it is green, the card is in standby mode. This is normal operation for the initial power-up. If needed, refer to the *Configuring* PAC/*PSC and Line Card Availability* section of the Configuring System Settings chapter in this reference for information on making the card active.

Ethernet 10/100 and Ethernet 1000/QGLC Standby LED States

The *Standby* LEDs on the Ethernet 10/100 and Ethernet 1000/QGLC Line Cards indicate that software is loaded on the cards, but are serving as redundant components.

Important: Quad Gigabit Ethernet line cards (QGLC) only work in an ASR 5000 behind a PSC.

The line cards installed will remain in a ready mode until their corresponding PAC/PSC is made active via configuration. While in ready mode, the Active LED should be off. After the PAC/PSC is made active, the line card installed in the upper-rear chassis slot behind the PAC/PSC will also be made active. The line card installed in the lower-rear chassis slot behind the PAC/PSC will also enter the standby mode.

The possible states for this LED are described below. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 43. Ethernet 10/100 and Ethernet 1000/QGLC Standby LED States

Color	Description	Troubleshooting
Green	Card is in redundant mode	None needed for line cards installed in slots 33 through 39 and 42 through 48 after configuration. If green for line cards installed in slots 17 through 23 and 26 through 32, refer to the *Monitoring the System* chapter of this reference for information on determining the status of the line card and system software processes and functionality.

Color	Description	Troubleshooting
None	Card in Ready ModeORCard is not receiving power OR Card in Active Mode	• This is normal prior to configuration. Neither the Active nor *Standby* LEDs on the card is on. • Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the *Ethernet 10/100 and Ethernet 1000/QGLC RUN/FAIL LED States* section of this chapter for troubleshooting information. • Check the state of the *Active* LED. If it is green, the card is in standby mode. If needed, refer to the *Manually Initiating a Line Card Switch* section in this chapter for information on configuring the card to serve as a redundant component.

Ethernet 10/100 and Ethernet 1000/QGLC Interface Link LED States

The *Link* LEDs, associated with a particular network interface on the Ethernet 10/100 and Ethernet 1000/QGLC Line Cards, indicate the status of the network link. These LEDs should be green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 44. Ethernet 10/100 and Ethernet 1000 Interface Link LED States

Color	Description	Troubleshooting
Green	Link is up	None needed. ***Important:*** This LED will not indicate the presence of a network link until the interface parameters are set during the software configuration process.
None	No power to card**OR**Link is down	• Verify that the RUN/FAIL LED is green. If so, the card is receiving power. If it is off, refer to *Ethernet 10/100 and Ethernet 1000/QGLC RUN/FAIL LED States* section for troubleshooting information. • Verify that the interface is cabled properly. • Verify that the device where the interface is connected to is cabled and powered properly. • Check the cable for continuity.

Ethernet 10/100 and Ethernet 1000/QGLC Interface Activity LED States

The *Activity* LEDs, associated with a particular network interface on the Ethernet 10/100 and Ethernet 1000/QGLC Line Cards, indicate the presence of traffic on the network link. These LEDs should be green when data is being transmitted or received over the interface.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 45. Ethernet 10/100 and Ethernet 1000 Interface Activity LED States

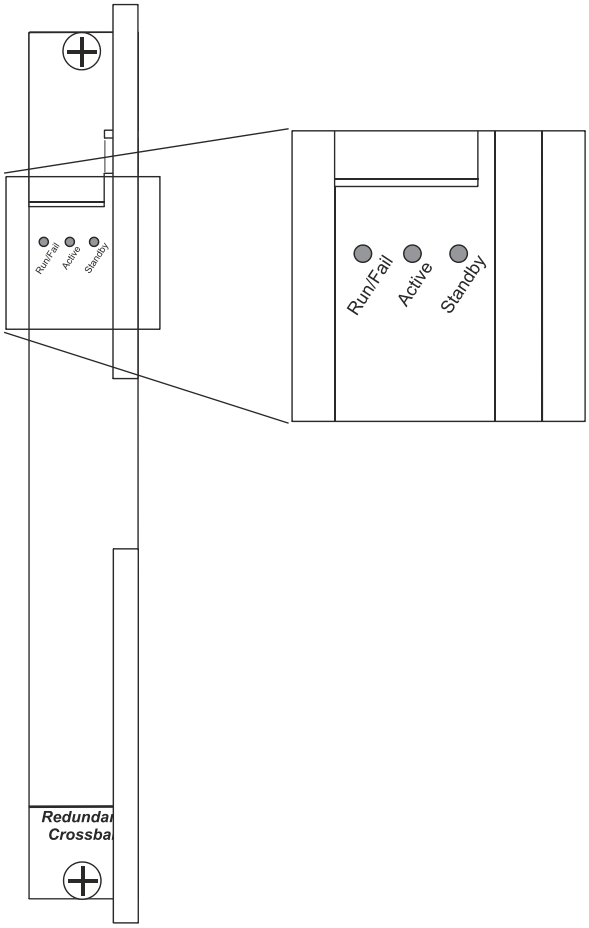
Color	Description	Troubleshooting
Flashing Green	Traffic is present on the link	None needed.
None	No traffic is present on the link	None needed if there is no activity on the link. Prior to configuration, this is normal operation.

Checking the LEDs on the RCC(s)

Each RCC is equipped with status LEDs as listed below:

- RUN/FAIL
- Active
- Standby

Figure 20. RCC LED Locations



The possible states for all of the SPIO's LEDs are described in the sections that follow.

RCC RUN/FAIL LED States

The RCC's *RUN/FAIL* LED indicates the overall status of the card. This LED should be green for normal operation.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 46. RCC RUN/FAIL LED States

Color	Description	Troubleshooting
Green	Card powered with no errors detected	None needed.

Color	Description	Troubleshooting
Red	Card powered with error(s) detected	Errors were detected during the POSTs. It is likely that the errors were logged to the system's command line interface during the boot process. Refer to the *Monitoring the System* chapter of this reference for information on determining the status of system hardware components.
None	Card powered with error(s) detected	• Verify that the *POWER* LEDs on the PFUs are green. If they are not, refer to the *Checking the LED on the PFU(s)* section in this chapter for troubleshooting information. • Verify that the power source is supplying ample voltage and current to the chassis. • Verify that the card is properly installed per the instructions in the *Hardware Installation and Administration Guide*. • If all of the above suggestions have been verified, it is possible that the SPIO is not functional. Please contact your service representative.

RCC Active LED States

The *Active* LED on the RCC indicates that the card is being used. For normal operation, this LED should be off on both RCCs.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 47. RCC Active LED States

Color	Description	Troubleshooting
Green	Card is active	The RCC is actively routing traffic from a line card installed behind a PAC/PSC that has failed to a redundant PAC/PSC. The RCC installed in chassis slot 40 processes traffic for the line cards in chassis slots 17 through 23 and 26 through 32. The RCC installed in chassis slot 41 processes traffic for the line cards in slots 33 through 39 and 42 through 48. Refer to either the *Checking the LEDs on the PAC(s)* or *Checking the LEDs on the PSC(s)* section of this chapter to determine which PAC/PSC has failed.
None	Card is not receiving power **OR** Card in Standby Mode	• Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the *RCC RUN/FAIL LED States* section of this chapter for troubleshooting information. • Check the state of the *Standby* LED. If it is green, the card is in standby mode. This is the normal operating mode.

RCC Standby LED States

The *Standby* LED on the RCC indicates that software is loaded on the card and is ready to provide a path for data or signalling traffic from a line card to a redundant PAC/PSC. This LED should be on for normal operation for both RCCs installed.

The possible states for this LED are described in the following table. If the LED is not green, use the troubleshooting information in the table to diagnose the problem.

Table 48. RCC Standby LED States

Color	Description	Troubleshooting
Green	Card is in standby mode	This is the normal operating mode.
None	Card is not receiving power **OR**Card in Active Mode	• Verify that the *RUN/FAIL* LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the *RCC RUN/FAIL LED States* section of this chapter for troubleshooting information. • Check the state of the *Active* LED. If it is green, the card is in active mode and the RCC is actively routing traffic from a line card installed behind a PAC/PSC that has failed. • Refer to either the *Checking the LEDs on the PAC(s)* or *Checking the LEDs on the PSC(s)* section of this chapter to determine which PAC/PSC has failed. Information on determining the cause of the failure can be found in the *Monitoring the System* chapter of this reference.

Testing System Alarm Outputs

The system provides the following two physical alarm mechanisms:

- **System Audible Alarm:** Located on the SPC/SMC, the speaker is used to provide an audible indicator that a minor, major, or critical alarm has occurred.
- **CO Alarms Interface:** Located on the SPIO, this interface provides a 10-pin connector that enables three normally-closed dry-contact relays for the triggering of external audio and/or visual indicators. These indicators can be used to alert that either a minor, major, or critical alarm has occurred.

The operation of these alarms can be tested by issuing the following command:

```
test alarm { audible | central-office [ critical | major | minor ] }
```

Keyword/Variable	Description
`audible`	Tests the CO Alarm Speaker on the SPC/SMC to verify operation.

Keyword/Variable	Description
`central-office`	Tests the CO Alarm Interface on the SPIO to verify operation. Individual alarms can be tested by using one of the following keywords: • **`critical`:** Specifies that the critical CO Alarms output is to be tested. • **`major`:** Specifies that the major CO Alarms output is to be tested. • **`minor`:** Specifies that the minor CO Alarms output is to be tested. If no keyword is specified, all alarms are tested.

When this command is executed, the specified alarm is activated for a period of 10 seconds. After this time, the alarm will return to its previous condition.

Taking Corrective Action

In the event that an issue was discovered with an installed application or line card, depending on the severity, it may be necessary to take corrective action.

The system provides several redundancy and fail-over mechanisms to address issues with application and line cards in order to minimize system downtime and data loss. These mechanisms are described in the sections that follow.

Manually Initiating a Management Card Switchover

When the system boots up, the SPC/SMC installed in chassis slot eight will boot into the “active” mode and begin booting other system components. The SPC/SMC installed in chassis slot nine will automatically be booted into “standby” mode dictating that it will serve as a redundant component. However, the active SPC/SMC will frequently synchronize currently running tasks or processes with the redundant SPC/SMC.

In the event of a critical failure on the SPC/SMC in slot eight, system control will be switched to the redundant SPC/SMC in slot nine. This is a relatively seamless transition because the two are synchronized. The formerly active SPC/SMC will then enter the standby mode allowing it to be safely replaced or restored.

In the event that an issue arises that is not severe enough for the system to perform an automatic switchover, a manual switch over can be invoked by executing the following instructions from the Exec mode prompt:

```
[local]host_name#
```

Step 1 Initiate a manual SPC/SMC switch over by entering one of the following commands:

For SPC:

```
card spc switchover
```

For SMC:

```
card smc switchover
```

You will receive the following prompt:

```
Are You Sure? [Yes|No]:
```

Step 2 Press **Y** to start the switchover.

Step 3 Verify that the switchover was successful by entering the following command at the prompt:

```
show card table
```

Check the entry in the *Oper State* column next to the SPC/SMC just switched. Its state should be *Standby*.

Manually Initiating a Packet Processing Card Migration

When the system boots up, all installed PACs/PSCs enter the "standby" mode. The standby mode indicates that the card is available for use but is not configured for operation. Installed components can be made active through the software configuration process. Cards that are not configured to enter the "active" mode will remain in standby mode for use as redundant components.

In the event of PAC/PSC critical failure, tasks will be automatically be migrated from the active card to a redundant card in standby mode. The line card installed behind the PAC/PSC that was formerly active will still be used to maintain the interfaces to external network equipment. Installed Redundancy Crossbar Cards (RCCs) will provide a path for signalling and data traffic between the line card and the now active PAC/PSC. Therefore, redundant PACs/PSCs do not require that line cards be installed behind them.

In the event that an issue arises that is not severe enough for the system to perform an automatic migration, a manual migration can be invoked. Follow the instructions below to manually initiate a PAC/PSC migration. These instructions assume you are at the root prompt for the Exec mode:

```
[local]host_name#
```

Step 1 Initiate a manual PAC/PSC migration by entering one of the following commands:

For PAC:

```
card pac migration from <original_slot#> to <final_slot#>
```

For PSC:

```
card psc migration from <original_slot#> to <final_slot#>
```

Keyword/Variable	Description
from	Specifies the chassis slot number of the PAC/PSC that is being migrated from. original_slot can be any value between 1 through 7, and 10 through 16.
to	Specifies the chassis slot number of the PAC/PSC that is being migrated to. Variable final_slot can be any value between 1 through 7, and 10 through 16.

You will receive the following prompt:

```
Are You Sure? [Yes|No]:
```

Step 2 Press **Y** to start the migration.

Step 3 Verify that the migration was successful by entering the following command at the prompt:

```
show card table
```

Check the entry in the *Oper State* column next to the PAC/PSC that was just migrated from. Its state should be *Standby*. The state of the PAC/PSC migrated to should be *Active*.

Manually Initiating a Line Card Switch

Line cards are installed in the half-height slots at the rear of the chassis. This design allows for two line cards (both Ethernet 10/100 both Ethernet 1000 (or both QGLCs on an ASR 5000)) to be installed behind every application card. When two line cards are installed, as the application card that they are installed behind is booted, the card in the upper-rear chassis slot will automatically be made active while the card in the lower-rear chassis slot will automatically be placed in standby mode. In the event that the active card experiences a failure, the system will automatically switch traffic to the standby card in the lower slot.

In the event that a SPIO experiences a failure, the system will automatically switch traffic to the redundant SPIO installed behind the redundant SPC/SMC.

In the event that an issue arises that is not severe enough for the system to perform an automatic switch, a manual switch can be performed. Follow the instructions below to manually initiate a line card or SPIO switch. These instructions assume you are at the root prompt for the Exec mode:

```
[local]host_name#
```

Step 1 Initiate a manual line card or SPIO migration by entering the following command:

```
card [ lc | spio ] switch [ to <slot#> ]
```

Keyword/Variable	Description
lc	Specifies that a switch will be done for an Ethernet 10/100 or Ethernet 1000 Line Card (Quad Gig-E (QGLC) on an ASR 5000 only).
spio	Specifies that a switch will be done for a SPIO.
to	Specifies the chassis slot number of the redundant Ethernet 10/100 or Ethernet 1000/QGLC Line Card that is being migrated to. Executing this command will switch network connections from the active line card that corresponds to the card being migrated to. *slot#* can be any of the following integer values: 17 through 23, 26 through 39, or 42 through 48. **Important:** This keyword is only used if the **lc** keyword is used. The line card being migrated to must be in the standby mode.

You will receive the following prompt:

```
Are You Sure? [Yes|No]:
```

Step 2 Press **Y** to start the switch.

Step 3 Verify that the migration was successful by entering the following command at the prompt:

```
show card table
```

Check the entry in the *Oper State* column next to the line card or SPIO that was just switched from. Its state should be *Standby*. The state of the line card or SPIO switched to should be *Active*.

Halting Cards

PACs/PSCs or line cards that are in either the active or standby modes can be halted. Halting these cards places them into the "offline" mode. When in this mode, the card will become unusable for session processing as either an active or redundant component.

If a card in the active mode is halted, its tasks, processes, or network connections will be migrated or switched to a redundant component prior to entering the offline mode.

This section provides information and instructions for initiating a card halt and restoring halted components.

Initiate a Card Halt

Follow the instructions below to manually initiate a card halt. These instructions assume you are at the root prompt for the Exec mode:

```
[local]host_name#
```

Step 1 Initiate a manual line card or SPIO migration by entering the following command:

```
card halt <slot#>
```

slot# is the chassis slot number in which the card to be halted is installed. It can be any integer value between 1 and 7, 10 through 48.

You will receive the following prompt:

```
Are You Sure? [Yes|No]:
```

Step 2 Press **Y** to start the halt of the card.

Step 3 Verify that the migration was successful by entering the following command at the prompt:

```
show card table
```

Check the entry in the *Oper State* column next to the line card that was just halted. Its state should be *Offline*. If the card was in active mode prior to the execution of this command, the state of the redundant component associated with it should now be *Active*.

Restoring a Previously Halted Card

Follow the instructions below to restore a card that was previously halted. These instructions assume you are at the root prompt for the Exec mode:

```
[local]host_name#
```

Step 1 Reboot the card to be restored by entering the following command:

```
card reboot <slot#> -force
```

You will receive the following prompt:

```
Are You Sure? [Yes|No]:
```

Step 2 Press **Y** to start the reboot of the card.

Step 3 Verify that the migration was successful by entering the following command at the prompt:

```
show card table
```

Check the entry in the *Oper State* column next to the line card that was just restored. Its state should be the state of that it was in before it was halted.

Verifying Network Connectivity

There are multiple commands supported by the system to verify and/or troubleshoot network connectivity. Note that network connectivity can only be tested once system interfaces and ports have been configured and bound.

The commands specified in this section should be issued on a context-by-context basis. This is due to the fact that contexts act like virtual private networks (VPNs) in that they operate independently of the others. Therefore, ports, interfaces, and routes configured in one context can not be tested from another without additional configuration.

To switch between contexts you must enter the following command at the root prompt for the Exec mode:

```
context <context_name>
```

context_name is the name of the context that you wish to switch to. The following prompt appears:

```
[context_name]host_name#
```

Using the Ping Command

Using the ping command verifies the system's ability to communicate with a remote node in the network by passing data packets between and measuring the response. This command is useful in verifying network routing and if a remote node is able to respond at the IP layer. The command has the following syntax:

```
ping <host_ip_address> [ count <num_packets> ] [ pattern <packet_pattern> ] [
size <octet_count> ] [ src { <src_host_name> | <src_host_ip_address> } ]
```

Keyword/Variable	Description
<host_ip_address>	Identifies the remote node to which is the target of the ping command. *host_ip_address* specifies the remote node using the node's assigned IP address specified using the standard IPv4.
count *<num_packets>*	Specifies the number of packets to send to the remote host for verification. *num_packets* must be within the range 1 through 10000. The default is 5.
pattern *<packet_pattern>*	Specifies a pattern to use to fill the internet control message protocol packets with. *packet_pattern* must be specified in hexadecimal format with a value in the range hexadecimal 0x0000 through 0xFFFF. *packet_pattern* must begin with a '0x' followed by up to 4 hexadecimal digits. The default is that each octet of the packet is encoded with the octet number of the packet.
size *<octet_count>*	Specifies the number of bytes each IP datagram. *octet_count* must be a value in the range 40 through 18432. The default is 56.

Keyword/Variable	Description
src { *<src_host_name>* | *<src_host_ip_address>* }	Specifies an IP address to use in the packets as the source node. *src_host_name*: specifies the source node using the node's logical host name which must be resolved via DNS lookup. *src_host_ip_address*: specifies the source node using the node's assigned IP address specified using the standard IPv4. The default is the IP address of the interface through which the ping was issued.

The following displays a sample of a successful response.

```
PING 192.168.250.1 (192.168.250.1): 56 data bytes

64 bytes from 192.168.250.1: icmp_seq=0 ttl=255 time=0.4 ms

64 bytes from 192.168.250.1: icmp_seq=1 ttl=255 time=0.2 ms

64 bytes from 192.168.250.1: icmp_seq=2 ttl=255 time=0.2 ms

64 bytes from 192.168.250.1: icmp_seq=3 ttl=255 time=0.2 ms

64 bytes from 192.168.250.1: icmp_seq=4 ttl=255 time=0.2 ms

--- 192.168.250.1 ping statistics ---

5 packets transmitted, 5 packets received, 0% packet loss

round-trip min/avg/max = 0.2/0.2/0.4 ms
```

If no response is received from the target follow these troubleshooting procedures:

- Verify that the correct IP address was entered.
- Attempt to ping a different device on the same network. If the ping was successful then it is likely that your system configuration is correct. Verify that the device you are attempting to ping is powered and functioning properly.
- Verify the port is operational.
- Verify that your configuration of the ports and interfaces within the context are correct.
- If your configuration is correct and you have access to the device that you're attempting to ping, ping the system from that device.
- If there is still no response, it is likely that the packets are getting discarded by a network device. Use the traceroute and show ip static-route commands discussed in this chapter to further troubleshoot the issue.

Using the Traceroute Command

The traceroute command collects information on the route data will take to a specified host. This is a useful troubleshooting command that can be used to identify the source of significant packet delays or packet loss on the network. This command can also be used to identify bottle necks in the routing of data over the network.

The command has the following syntax:

```
traceroute { <host_name> | <host_ip_address> } [ count <packets> ] [ df ]
[ maxttl <max_ttl> ] [ minttl <min_ttl> ] [ port <port_number> ] [ size
<octet_count> ] [ src { <src_host_name> | <src_host_ip_address> } ] [
timeout <seconds> ]
```

Keyword/Variable	Description
<host_name>	Identifies the remote node to trace the route to by the hostname. *host_name* specifies the remote node using the node's logical host name which must be resolved via DNS lookup.
<host_ip_address>	Identifies the remote node to trace the route to by the IP address. *host_ip_address* specifies the remote node using the node's assigned IP address specified using the standard IPv4.
count	Specifies the number of UDP probe packets to send. The default is 3.
df	Indicates the packets for the tracing of the route should not be fragmented. If a packet would require fragmenting then it is dropped and the ICMP response "Unreachable, Needs Fragmentation" is received.
maxttl *<max_ttl>*	Specifies the maximum time to live (TTL), in seconds, for the route tracing packets. *max_ttl* must be specified as a value in the range of 1 through 255. It is an error if *max_ttl* is less than *min_ttl*, whether *min_ttl* is specified or defaulted. The time to live is the number of hops through the network, i.e., it is not a measure of time. The default maximum TTL is 30 seconds.
minttl *<min_ttl>*	Specifies the minimum time to live, in seconds, for the route tracing packets. *min_ttl* must be specified as a value in the range of 1 through 255. It is an error if *min_ttl* is greater than *max_ttl*, whether *max_ttl* is specified or defaulted. The time to live is the number of hops through the network, i.e., it is not a measure of time. The default minimum TTL is 1 second.
port *<port_number>*	Specifies a specific port to connect to where *port_number* must be a value in the range 1 through 65535. The default port is 33434.
size	Specifies the number of bytes each packet. *octet_count* must be a value in the range 40 through 32768. The default is 40.
src { *<src_host_name>* | *<src_host_ip_address>* }	Specifies an IP address to use in the packets as the source node. *src_host_name*: specifies the remote node using the node's logical host name which must be resolved via DNS lookup. *src_host_ip_address*: specifies the remote node using the node's assigned IP address specified using the standard IPv4. The default is the IP address of the interface through which the ping was issued.
timeout *<seconds>*	Specifies the maximum time to wait for a response from each route tracing packet. *seconds* must be a value in the range 2 through 100. The default is 5.

The following displays a sample output.

```
traceroute to 192.168.250.1 (192.168.250.1), 30 hops max, 40 byte packets
```

```
1 192.168.250.1 (192.168.250.1) 0.446 ms 0.235 ms 0.178 ms
```

Viewing IP Routes

The system provides a mechanism for viewing route information to a specific node or for an entire context. This information can be used to verify network connectivity and to ensure the efficiency of the network connection. The command has the following syntax:

show ip route [*<route_ip_address>* [*<route_gw_address>*]]

Keyword/Variable	Description
<route_ip_address>	Specifies the IP address of a network node for which route information is displayed.
<route_gw_address>	Specifies the IP address of the gateway router between the system and the network node for which route information is displayed. This is an optional keyword.

If no keywords are specified, all IP routes within the context's routing table are displayed.

The following displays a sample of this command's output showing a context routing table.

```
"*" indicates the Best or Used route.

 Destination Nexthop Protocol Prec Cost Interface

*0.0.0.0/0 10.0.4.1 static 0 0 SPIO1

*10.0.4.0/24 0.0.0.0 kernel 0 0 SPIO1

*10.0.4.0/32 0.0.0.0 kernel 0 0 SPIO1

*10.0.4.3/32 0.0.0.0 kernel 0 0 SPIO1

*10.0.4.255/32 0.0.0.0 kernel 0 0 SPIO1
```

Viewing the Address Resolution Protocol Table

The system provides a mechanism for viewing Address Resolution Protocol (ARP) table information to a specific node or for an entire context. This information can be used to verify that when the system sends an ARP packet, it receives valid responses from other network nodes. The command has the following syntax:

show ip arp [*<arp_ip_address>*]

arp_ip_address specifies a specific network node for which to display ARP information. If this keyword is not specified, all entries within the context's ARP table are displayed.

Important: When the VPN Manager restarts, it removes all interfaces from the kernel and thus the kernel removes all ARP entries. When this happens, the NPU still holds all of the ARP entries so that there is no traffic disruption. When this happens, from a user point of view, **show ip arp** is broken since this command gathers information from the kernel and not the NPU.

The following displays a sample of this command's output showing a context's ARP table.

```
Flags codes:

C - Completed, M - Permanent, P - Published, ! - Not answered

T - has requested trailers

Address Link Type Link Address Flags Mask Interface

10.0.4.240 ether 00:05:47:02:20:20 C SPIO1

10.0.4.7 ether 00:05:47:02:03:36 C SPIO1

10.0.4.1 ether 00:01:30:F2:7F:00 C SPIO1
```

Using the System Diagnostic Utilities

The system provides protocol monitor and test utilities that can are useful when troubleshooting or verifying configurations. The information generated by these utilities can in many cases either identify the root cause of a software or network configuration issue or, at the very least, greatly reduce the number of possibilities.

This section contains information and instructions for using these utilities.

Using the Monitor Utility

For troubleshooting purposes, the system provides a powerful protocol monitoring utility. This tool can be used to display protocol information for a particular subscriber session or for every session being processed.

Caution: The monitor tool is intrusive in that it may cause session processing delays and/or data loss. Therefore, it should be used only when troubleshooting .

Using the Protocol Monitor

The system's protocol monitor displays information for every session that is currently being being processed. Depending on the number of protocols monitored, and the number of sessions in progress, a significant amount of data is generated. It is highly recommended that logging be enabled on your terminal client in order to capture all of the information that is generated.

Follow the instructions in this section to invoke and configure the protocol monitoring tool.

Step 1 Invoke the protocol monitor by entering the following command:

```
monitor protocol
```

The following output is displayed.

```
MONITOR GLOBAL PROTOCOLS:
 11 - SNMP 21 - L2TP (Admin only)
 12 - RADIUS Authentication (Admin only) 22 - L2TPMGR (Admin only)
 13 - RADIUS Accounting (Admin only) 23 - L2TP Data(Admin only)
```

```
14 - A11 (R-P Interface) (Admin only) 24 - GTP (Admin only)
15 - Mobile IPv4 (Admin only) 25 - GTPCMGR (Admin only)
16 - A11MGR (Admin only) 26 - GTPU (Admin only)
17 - PPP (Admin only) 27 - GTPP (Admin only)
18 - A10 (Admin only) 28 - DHCP (Admin only)
19 - User L3 (Admin only) 29 - CDR (Admin only)
31 - RADIUS COA (Admin only) 30 - DHCPV6 (Admin only)
51 - SCTP (Admin only)
32 - MIP Tunnel (Admin only) 52 - M3UA (Admin only)
33 - L3 Tunnel (Admin only) 53 - SCCP (Admin only)
34 - CSS Data (Admin only) 54 - TCAP (Admin only)
35 - CSS Signaling (Admin only) 55 - MAP (Admin only)
36 - EC Diameter (Admin only) 56 - RANAP (Admin only)
37 - SIP (IMS) (Admin only) 57 - GMM (Admin only)
38 - IPSec IKE Only (Admin only) 58 - GPRS-NS (Admin only)
59 - BSSGP (Admin only)
40 - IPSec IKEv2 Only (Admin only)
41 - IPSG RADIUS Signal (Admin only) 61 - SSCOP (Admin only)
42 - ROHC (Admin only) 62 - SSCFNNI (Admin only)
43 - WiMAX R6 (Admin only) 63 - MTP3 (Admin only)
44 - WiMAX Data (Admin only) 64 - LLC (Admin only)
45 - SRP (Admin only) 65 - SNDCP (Admin only)
46 - BCMCS SERV AUTH (Admin only) 66 - BSSAP+ (Admin only)
47 - RSVP (Admin only) 67 - SMS (Admin only)
48 - Mobile IPv6 (Admin only) 68 - MTP2 (Admin only)
49 - ASNGWMGR (Admin only)
50 - STUN                    (Admin only)
70 - DNS Client (Admin only)
75 - App Specific Diameter (Admin only)
```

```
   81 - S1AP                    (Admin only)   82 - NAS (Admin
only)

 (B)egin Protocol Decoding, (Q)uit, <ESC> Prev Menu

 Select:
```

Step 2 Choose the protocol that you wish to monitor by entering the number associated with the protocol (11 through 19 and 21 through 28) at the *Select*: prompt. A greater-than sign (>) appears next to the protocol you selected.

Step 3 Repeat *step 2* as needed to choose multiple protocols.

Step 4 Press **B** to begin the protocol monitor. If you selected any protocol other than 11 (SNMP), the following message is displayed:

```
WARNING!!! You have selected options that can DISRUPT USER SERVICE

 Existing CALLS MAY BE DROPPED and/or new CALLS MAY FAIL!!!

 (Under heavy call load, some debugging output may not be
displayed)

Proceed? - Select (Y)es or (N)o
```

Step 5 Enter **Y** to proceed with the monitor or **N** to go back to the previous menu.

```
C - Control Events (ON )

D - Data Events (ON )

E - EventID Info (ON )

I - Inbound Events (ON )

O - Outbound Events (ON )

S - Sender Info (OFF)

T - Timestamps (ON )

X - PDU Hexdump (OFF)

A - PDU Hex/Ascii (OFF)

+/- Verbosity Level ( 1)

L - Limit Context (OFF)

M - Match Newcalls (ON )

R - RADIUS Dict (no-override)

G - GTPP Dict (no-override)
```

```
Q)uit, <ENTER> Display Options, <ESC> Prev Menu, <SPACE> Pause
```

Step 6 Configure the amount of information that is displayed by the monitor. To enable or disable options, enter the letter associated with that option (C, D, E, etc.). To increase or decrease the verbosity, use the plus (+) or minus (-) keys.

The current state, ON (enabled) or OFF (disabled), is shown to the right of each option.

Step 7 Press the **Enter** key to refresh the screen and begin monitoring.

The monitor remains active until disabled. To quit the protocol monitor and return to the prompt, press **q**.

Using the Protocol Monitor for a Specific Subscriber

The system's protocol monitor can be used to display information for a specific subscriber session that is currently being processed. Depending on the number of protocols monitored, and the number of sessions in progress, a significant amount of data is generated. It is highly recommended that logging be enabled on your terminal client in order to capture all of the information that is generated.

Follow the instructions in this section to invoke and configure the protocol monitoring tool for a specific subscriber session.

Step 1 To invoke the session-specific protocol monitor enter the following command:

```
monitor subscriber
```

The following screen is displayed, followed by a list of all available monitoring methods:

```
MONITOR SUBSCRIBER:

1) By MSID/IMSI

2) By Username

3) By Callid

4) By IP Address

5) By IPv6 Address

6) Next-Call

7) Next-BCMCS-Call

8) Next-BCMCS-Service-Request

9) By IMEI

a) By MSISDN

b) Next-1xRTT Call

c) Next-ASNGW Call
```

```
 d) Next-CLOSEDRP Call

 e) Next-EVDO-Rev0 Call

 f) Next-EVDO-RevA Call

 g) Next-GGSN Call

 h) Next-HA Call

 i) Next-IPSG Call

 j) Next-LNS Call

 k) Next-PDIF Call

 l) By ASN Peer IP Address

 m) By PDIF Peer IP Address

 n) By Peer LAC IP Address

 o) By SGSN IP Address

 p) By PCF IP Address

 r) By Peer FA IP Address

 s) By IPSG Peer IP Address

 t) Next-ASNPC Call

 u) Next-OpenRP Call

 v) Next-Call By APN


 Q) Quit


<ESC> Return to Previous Menu
```

Step 2 Specify the method the monitor should use to select the user to monitor by entering the number that corresponds with the desired selection in the menu.

Step 3 Enter the appropriate information for the menu item selected.

If no session matching the specified criteria was being processed when the monitor was invoked, the following output is displayed:

```
NO MATCHING CALL - waiting for a matching call to connect...
```

```
C - Control Events (ON ) 11 - PPP (ON ) 21 - L2TP (ON )

D - Data Events (ON ) 12 - A11 (ON ) 22 - L2TPMGR (OFF)

E - EventID Info (ON ) 13 - RADIUS Auth (ON ) 23 - L2TP Data (OFF)

I - Inbound Events (ON ) 14 - RADIUS Acct (ON ) 24 - GTPC (ON )

O - Outbound Events (ON ) 15 - Mobile IPv4 (ON ) 25 - GTPCMGR
(OFF)

S - Sender Info (OFF) 16 - A11MGR (OFF) 26 - GTPU (OFF)

T - Timestamps (ON ) 17 - SESSMGR (ON ) 27 - GTPP (ON )

X - PDU Hexdump (OFF) 18 - A10 (OFF) 28 - DHCP (ON )

A - PDU Hex/Ascii (OFF) 19 - User L3 (OFF) 29 - CDR (ON )

+/- Verbosity Level ( 1) 31 - Radius COA (ON ) 30 - DHCPV6 (ON )

L - Limit Context (OFF) 32 - MIP Tunnel (ON ) 53 - SCCP (OFF)

M - Match Newcalls (ON ) 33 - L3 Tunnel (OFF) 54 - TCAP (OFF)

R - RADIUS Dict: (no-override) 34 - CSS Data (OFF) 55 - MAP (ON )

G - GTPP Dict: (no-override) 35 - CSS Signal (OFF) 56 - RANAP
(OFF)

Y - Multi-Call Trace (OFF) 36 - EC Diameter (ON ) 57 - GMM (ON )

 37 - SIP (IMS) (OFF) 58 - GPRS-NS (OFF)

 40 - IPSec IKEv2 (OFF) 59 - BSSGP (OFF)

 41 - IPSG RADIUS (ON ) 64 - LLC (OFF)

 42 - ROHC (OFF) 65 - SNDCP (OFF)

 43 - WiMAX R6 (ON ) 66 - BSSAP+ (OFF)

 44 - WiMAX Data (OFF) 67 - SMS (OFF)

 45 - SRP (OFF)

 46 - BCMCS SERV AUTH (OFF)

 47 - RSVP (ON )

 48 - Mobile IPv6 (ON )

 49 - ASNGWMGR (OFF)

 50 - STUN (IMS) (OFF)

 75 - App Specific Diameter OFF
```

```
 (Q)uit, <ESC> Prev Menu, <SPACE> Pause, <ENTER> Re-Display
Options
```

Step 4 Configure the amount of information that is displayed by the monitor. To enable or disable options, enter the letter associated with that option (C, D, E, etc.). To increase or decrease the verbosity, use the plus (+) or minus (-) keys.

The current state, ON (enabled) or OFF (disabled), is shown to the right of each option.

Important: Option **Y** for performing multi-call traces is only supported for use with the GGSN.

Step 5 Repeat *step 6* as needed to enable or disable multiple protocols.

Step 6 Press the **Enter** key to refresh the screen and begin monitoring.

The following displays a portion of a sample of the monitor's output for a subscriber named user2@aaa. The default protocols were monitored.

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

Incoming Call:

-------------------------------------------------------------------
---------

 MSID: 0000012345 Callid: 002dc6c2

 Username: user2@aaa SessionType: unknown

 Status: Active Service Name: xxx1

 Src Context: source Dest Context:

-------------------------------------------------------------------
---------


<<<<OUTBOUND 10:02:35:415 Eventid:25001(0)

PPP Tx PDU (9)

PAP 9: Auth-Ack(1), Msg=


<<<<OUTBOUND 10:02:35:416 Eventid:25001(0)
```

```
PPP Tx PDU (14)

IPCP 14: Conf-Req(1), IP-Addr=192.168.250.70


<<<<OUTBOUND 10:02:35:416 Eventid:25001(0)

PPP Tx PDU (27)

CCP 27: Conf-Req(1), MPPC, Stac-LZS, Deflate, MVRCA


INBOUND>>>>> 10:02:35:517 Eventid:25000(0)

PPP Rx PDU (30)

IPCP 30: Conf-Req(1), IP-Comp VJ-Comp, IP-Addr=0.0.0.0, Pri-
DNS=0.0.0.0,

Sec-DNS=0.0.0.0


<<<<OUTBOUND 10:02:35:517 Eventid:25001(0)

PPP Tx PDU (26)

IPCP 26: Conf-Rej(1), IP-Comp VJ-Comp, Pri-DNS=0.0.0.0, Sec-
DNS=0.0.0.0


INBOUND>>>>> 10:02:35:517 Eventid:25000(0)

PPP Rx PDU (12)

IPCP 12: Conf-Ack(1), IP-Addr=192.168.250.70


INBOUND>>>>> 10:02:35:518 Eventid:25000(0)

PPP Rx PDU (31)

LCP 31: Prot-Rej(1), Rejected-Protocol=CCP (0x80fd)


INBOUND>>>>> 10:02:35:518 Eventid:25000(0)

PPP Rx PDU (12)

IPCP 12: Conf-Req(2), IP-Addr=0.0.0.0
```

```
<<<<OUTBOUND 10:02:35:518 Eventid:25001(0)

PPP Tx PDU (14)

IPCP 14: Conf-Nak(2), IP-Addr=192.168.250.87


INBOUND>>>>> 10:02:35:519 Eventid:25000(0)

PPP Rx PDU (12)

IPCP 12: Conf-Req(3), IP-Addr=192.168.250.87
```

The monitor remains active until disabled. To quit the protocol monitor and return to the prompt, press **q**.

Using the DHCP Testing Tool

The CLI provides a mechanism for testing network connectivity with and configuration of DHCP servers. This functionality can be extremely useful in determining the accuracy of the system's DHCP configuration and troubleshooting the server's response time.

This tool provides a mechanism from getting an IP address for one or more DHCP servers that the system is configured to communicate with. In addition, the tool provides a mechanism for releasing the address back to the DHCP server.

Important: This tool must be executed from the context in which the DHCP server(s) are configured.

To execute the DHCP test tool enter the following command:

dhcp test dhcp-service { *<service_name>* } [**all** | **<server** *<ip_addr>*|

Table 49. Sample dhcp test dhcp-service Command Output

Keyword	Description
<service_name>	Name of DHCP service from which to lease an IP address.
all	Test all DHCP servers in this DHCP service.
server	Followed by IP address of server under test.

Appendix A
Engineering Rules

This section provides engineering rules or guidelines to consider prior to configuring the system for your network deployment.

CLI Rules

Multiple CLI session support is based on the amount of available memory. The Resource Manager reserves enough resources so that at a minimum, support for six CLI sessions is assured at all times. One of the six sessions is further reserved for use exclusively by a CLI session on an SPIO serial interface.

Additional CLI sessions beyond the pre-reserved limit are permitted if sufficient SMC or SPC resources are available. If the Resource Manager is unable to reserve resources for a CLI session beyond those that are pre-reserved, users with administrator-privileges are prompted to create the new CLI session, even without reserved resources.

Interface and Port Rules

The rules discussed in this section pertain to the following Ethernet line cards and the type of interfaces they facilitate, regardless of the application:

- Fast Ethernet 10/100 (FELC)
- Gigabit Ethernet 1000 (GELC)
- Quad Gigabit Ethernet (QGLC)

Line Card Rules

The following engineering rules apply to the Fast Ethernet 10/100, Gigabit Ethernet 1000, and Quad Gigabit line cards:

- Give all logical interfaces a unique name to identify the interface from others in the same context. Logical interfaces in different contexts may have the same name.
- A single physical port can support multiple logical interfaces when you configure VLAN tags for that physical port. You can use VLAN tagging to bind a single physical port to multiple logical interfaces that reside in different contexts.
- Assign all logical interfaces a valid IP address and subnet.
    - Give each logical interface within a context a unique IP address(es). Logical interfaces in different contexts can have the same IP address(es).
    - If multi-homing is supported on the network, you can assign all logical interfaces a single primary IP address and up to 16 secondary IP addresses.
- You can configure a logical interface in only one context, but you can configure multiple interfaces (up to 512 Ethernet or 1,024 ATM) in a single context.
- You can apply a maximum of 128 ACL rules to a single logical interface.
- All ports are identified by their <slot#>/<port#>.
- Each physical port on a Gigabit Ethernet 1000 or Quad Gigabit line card may contain up to a maximum of 1024 VLAN tags.
- Each physical port on an Fast Ethernet 10/100 Line card may contain up to a maximum of 256 VLAN tags.
- The total number of VLANs untagged and/or tagged on each Fast Ethernet 10/100 Line Card must not exceed 1025 (8 untagged + 1017 tagged).
- A logical interface is limited to using a single VLAN or ATM PVC on a single physical port, identified by its <cardslot#/port#>.
- When using redundant (standby) line cards:

- You must configure the active line card only. In the event of a failover, all relevant information (such as the IP address) is transferred to the standby line card.
- Assure that the line cards installed in the upper and lower chassis slots behind a single PAC, PSC or PSC2 must be of the same type: Fast Ethernet 10/100, Gigabit Ethernet 1000, or Quad Gigabit line cards.

Important: If you have enabled the Port Redundancy feature, it is possible for ports on both line cards to be active while one provides line card redundancy for the other. With the port redundancy feature, each physical port has a primary MAC address. Each corresponding standby port has a different (alternate) MAC address.

Packet Data Network (PDN) Interface Rules

The following engineering rules apply to the interface to the packet data network (PDN):

- Configure the logical interfaces used to facilitate the PDN interface within the egress context.
- The default is to use a single interface within the egress context to facilitate the PDN interface.
- You can configure multiple interfaces in the egress context by using static routes or dynamic routing protocols.
- You may also configure next-hop default gateways.

Packet Processing Card Rules

The following engineering rules apply to the PAC, PSC, and PSC2 processing application cards:

- When you configure a processing line card to enter the active mode, it results in the following:
  - The total number of PACs, PSCs or PSC2s that will become operationally active is increased by one.
  - In the event of a failure, the line card(s) directly behind the PACs, PSC or PSC2 will become available directly, or to another PAC/PSC/PSC2 via the RCC.
- If you want processing-only application cards, all line card slots directly behind them can be empty. Otherwise, disable those line card slots with the **`shutdown`** command as described in the *Command Line Interface Reference*.
- If you want standby (redundant) PACs, PSCs or PSC2s, do not populate all line card slots directly behind them since they will not be used. If the slots are populated, disable the line card slots with the **`shutdown`** command as described in the *Command Line Interface Reference*.
- A line cards is not used unless the processing application card directly in front of it is made active.

Context Rules

- A maximum of 64 contexts may be configured per chassis.
- Up to 2000 IP address pools can be configured within a single context (regardless of the number of PACs, PSCs or PSC2s) with a total system limit of 5000 IP address pools for all contexts. However, there is also a limit of 4,000,000 addresses and 32,000,000 static addresses that can be configured per context. Therefore, the number of pools depends on how many addresses are being used and how they are subnetted.
- Each context supports up to 32,000,000 static IP pool addresses. You can configure a maximum total of M64 x 32M static IP pool addresses per chassis. Each static IP pool can contain up to 500,000 addresses.
- Each context supports up to 4,000,000 dynamic IP pool addresses. You can configure a maximum total of 4 x 4M dynamic IP pool addresses per chassis. Each dynamic IP pool can contain up to 500,000 addresses.

> ***Important:*** Each address in the pool requires approximately 60 bytes of memory. The amount of memory required, however, depends on a number of factors such as the pool type, and hold-timer usage. Therefore, in order to conserve available memory, you may need to limit the number of pools depending on the number of addresses to be configured and the number of installed application cards.

- A maximum of 1,000,000 IP addresses per chassis can be in use at any given time. The maximum number of simultaneous subscriber sessions is controlled by the installed capacity license for the service(s) supported.
- The maximum number of static routes that can be configured per context is 1200.
- The maximum number of static ARP entries per context is 128.
- The maximum number of domains per context is 2,048.
- ASN GW services configured within the same context cannot communicate with each other.
- A maximum of 512 BGP/AAA monitors can be configured per context for Interchassis Session Recovery support.
- You can configure up to 128 AAA servers per context for a default AAA server group. The servers can be configured as accounting, authentication, charging servers, or any combination thereof.
- You can configure up to 800 AAA server groups per context with following limitations:
    - Configure up to 128 servers per AAA server group. The servers can be configured as accounting, authentication, or charging servers, or any combination thereof.
    - Configure up to 1600 servers per context in AAA Server group mode. The servers can be configured as accounting, authentication, or charging servers, or any combination thereof.
    - Up to 800 NAS-IP address/NAS identifier (1 primary and 1 secondary per server group) can be configured per context.
- Up to 12 charging gateway functions (CGFs) for GTPP accounting can be configured per context.

Subscriber Rules

The following engineering rules apply to subscribers configured within the system:

- Configure a maximum of 2,048 local subscribers per context.
- You may configure attributes for each local subscriber.
- The system creates a default subscriber default for each context when the context is made. Configure attributes for each default subscriber. If a AAA-based subscriber is missing attributes in the authentication reply message, the default subscriber attributes in the context where the subscriber was authenticated are used.

  Important: Default is not used when local authentication (for local subscribers) is performed.

- Configure default subscriber templates on a per AAA realm (domain aliases configured within a context) basis.
- Configure default subscriber templates on a per PDSN, FA,ASN GW, or HA service.
- For AAA authenticated subscribers, the selection of local subscriber template to use for setting attributes is in the following order:
  - If the username (NAI) matches any local domain name and the domain name has a local subscriber name configured, that local subscriber template is used.
  - If the first case fails, and if the serving service has a default username configured, that subscriber template is used.
  - If the first two cases fail, the default subscriber template in the AAA context is used.

Service Rules

The following engineering rules apply to services configured within the system:

- Configure a maximum of 256 services (regardless of type) per system.

  Caution: Large numbers of services greatly increase the complexity of management and may affect overall system performance. Therefore, it is recommended that you do not configure a large number of services unless your application absolutely requires it. Please contact your local service representative for more information.

- The total number of entries per table and per chassis is limited to 256.
- Although you can use service names that are identical to those configured in different contexts on the same system, this is not a good practice. Services with the same name can lead to confusion and difficulty in troubleshooting problems, and make it difficult to understand the output of show commands.

Access Control List (ACL) Engineering Rules

The following rules apply to Access Control Lists:

- The maximum number of rules per ACL is 128.
- The maximum number of ACL rules applied per port is 128.
- The maximum number of ACL rules applied per context is 1024.
- The maximum number of ACL rules per IPSEC policy is 1.
    - The maximum number of IPSEC ACL rules per context is 1024.
- The maximum number of ACLs you can configure per context is limited by the number of rules allowed within each ACL. If each ACL contained the maximum number of rules (128), the maximum number of ACLs per context is 8 (128 rules per ACL multiplied by 8 ACLs = 1024 (the ACL rules limit per context)).
- The maximum number of ACLs applied to an IP access group is 1 whether it is configured for a port or context. Since the maximum number of IP access groups you can apply to an interface or context is 16, the following calculations apply:
    - For each interface/port: 8 rules per ACL multiplied by 16 IP access groups = 128 (the ACL rules limit per port)
    - For each context: 64 rules per ACL multiplied by 16 IP access groups = 1024 (the ACL rules limit per context)

Appendix B
System Software Task and Subsystem Descriptions

For redundancy, scalability and robust call processing, the system's software is divided into a series of tasks that perform specific functions. These tasks communicate with each other as needed to share control and data signals. As a result, system processes can be distributed across multiple tasks thus reducing the overall work-load on any given task and improving system performance. In addition, this distributed design provides fault containment that greatly minimizes the impact to the number of processes or sessions due to a failure.

All tasks run in a Common Firmware Environment (CFE) that resides on specialized Central Processing Units (CPUs) on each of the application cards. The Switch Processor Card (SPC) and System Management Card (SMC) each have a single CPU that is responsible for running tasks related to system management and control. Each Packet Accelerator Card (PAC) contains four CPUs (CPU 0 through 3, CPU 0 is the lead CPU). The Packet Services Card (PSC) contains two CPUs (CPU 0 and 1, CPU 0 is the lead CPU). The CPUs on the PAC and PSC are responsible for session processing, and for running the various tasks and processes required to handle the mobile data call. In addition to the CPUs, PACs and PSCs each have a Network Processor Unit (NPU) for IP forwarding.

The following sections describe the primary tasks that are implemented by the system:

Primary Task Subsystems

The individual tasks that run on the CPUs are divided into subsystems. Following is a list of the primary subsystems responsible for call session processing:

- **System Initiation Task (SIT):** This subsystem starts tasks and initializes the system. This includes starting a set of initial tasks at system startup time (static tasks), and starting individual tasks on demand at arbitrary times (dynamic tasks).
- **High Availability Task (HAT): W**ith the Recovery Control Task (RCT) subsystem, the HAT subsystem maintains the operational state of the system. HAT monitors the various software and hardware components of the system. If there are unusual activities, such as the unexpected termination of another task, the HAT subsystem takes a suitable course of action, such as triggering an event to the RCT subsystem to take corrective action or to report the status. The end result is that there is minimal or no impact to the service.
- **Recovery Control Task (RCT):** This subsystem executes a recovery action for any failure that occurs in the system. The RCT subsystem receives signals from the HAT subsystem (and in some cases from the NPU subsystem) and determines what recovery actions are needed.

  The RCT subsystem runs on the active SPC/SMC and synchronizes the information it contains with the RCT subsystem on the standby SPC/SMC.
- **Shared Configuration Task (SCT):** This subsystem provides a facility to set, retrieve, and receive notification of system configuration parameters. The SCT is mainly responsible for storing configuration data for the applications that run on the system.

  The SCT subsystem runs only on the active SPC/SMC and synchronizes the information it contains with the SCT subsystem on the standby SPC/SMC.
- **Resource Management (RM):** This subsystem assigns resources, such as CPU loading and memory, for every system task upon start-up. The RM subsystem monitors resource use to verify that allocations are as specified. RM also monitors all sessions and communicates with the Session Controller to enforce capacity licensing limits.
- **Virtual Private Network (VPN):** This subsystem manages the administrative and operational aspects of all VPN-related entities in the system. The functions performed by the VPN subsystem include:
  - Creating separate VPN contexts
  - Starting the IP services within a VPN context
  - Managing IP pools and subscriber IP addresses, and distributing the IP flow information within a VPN context.

  All IP operations within the system are done within specific VPN contexts. In general, packets are not forwarded across different VPN contexts. The only exception currently is the Session subsystem.
- **Network Processing Unit (NPU):** This subsystem is responsible for the following:
  - Using the database to match address and port numbers to destination tasks for fast-path forwarding of dataframes
  - Receiving and transmitting user data frames to/from various physical interfaces
  - IPv4 forwarding decisions (both unicast and multicast)
  - Per-interface packet filtering

- Traffic management and traffic engineering
- Passing user data frames to/from PAC/PSC CPUs
- Modifying, adding, or stripping datalink/network layer headers
- Recalculating checksums
- Maintaining statistics
- Managing external Ethernet interfaces

- **Card/Slot/Port (CSP):** Coordinates the events that occur when any card (application or line) is inserted, locked, unlocked, removed, shutdown, or migrated. SCP also performs auto-discovery and configures ports on a newly-inserted line card. It determines how line cards map to PAC or PSC/PSC2 cards (through a Redundancy Crossbar Card (RCC), if necessary).

  The CSP subsystem runs only on the active SPC/SMC and synchronizes the information it contains with the SCT subsystem on the standby SPC/SMC. It is started by the SIT subsystem and monitored by the HAT subsystem.

- **Session:** Performs high-touch processing of mobile subscribers' packet-oriented data session flows. High-touch user data processing consists of the following:
  - Payload transformation
  - Filtering and scheduling
  - Statistics collection
  - Policing

Primary Subsystem Composition

Many of the primary subsystems are composed of critical tasks—controller tasks called Controllers, and subordinated tasks called Managers. Critical tasks are essential to the system's ability to process calls, such as those in the SIT subsystem.

Controllers serve several purposes:

- They monitor the state of their Managers and allow communication between Managers within the same subsystem.
- They enable inter-subsystem communication since they can communicate with the controllers of other subsystems.
- They masks the distributed nature of the software from the user allowing for ease of management.

Managers manage resources and mappings between resources. In addition, some managers are directly responsible for call processing.

The following sections provide information about the composition of the primary subsystems that are composed of critical, controller, and /or manager tasks:

ST16 Subsystems

The following tables describe managers and tasks performing within the specified subsystems on an ST16.

Important: Variations regarding how the managers and tasks are distributed based on session recovery are included in the Card and CPU columns in some tables. Tables without these indicators are applicable to ST16s with and without session recovery. The ST16 dynamically distributes processes, tasks, and managers on startup. The following tables list the typical locations but variations can occur depending on available resources.

Table 50. ST16 System Initiation Subsystem

Task	Description	Card	CPU
SITMAIN	Initiated at system start-up, the primary Session Initiation Task (SITMAIN) performs the following functions: • Reads and provides startup configuration to other SIT components • Starts SITREAP sub-function • Maintains CPU state information	All	All

Task	Description	Card	CPU
SITPARENT sub-function	Initiated at system start-up, the parent Session Initiation Task (SITPARENT) performs the following functions: • Starts SPCs/SMCs in either active or standby mode • Registers tasks with HAT task • Notifies CSP task of CPU startup completion • Brings PACs up in standby mode **Important:** SITPARENT replaces the sub-function SITPAC.	All	All
SITREAP sub-function	Shuts down tasks as required	All	All

Table 51. ST16 High Availability Subsystem

Task	Description	Card	CPU
HAT System Controller (HATSYSTEM)	This is the main HAT task that controls all the HAT sub-function tasks in the system. It is initiated on system start-up and performs the following functions: • Initializes system components (such as the Gigabit Ethernet switches and switch fabric) • Monitors system components such as fans for state changes • Triggers actions for redundancy in the event of fault detection The HAT subsystem on the redundant SPC/SMC mirrors the HAT subsystem on the active SPC.	SPCs	0
HATCPU	• Performs device initialization and control functions because of the CPUs hardware capabilities • Reports the loss of any task on its CPU to HATSYSTEM sub-function • Controls the LEDs on the PAC • Initializes and monitors the dedicated hardware on PACs	All PACs	0
	• Collects CPU monitoring information periodically and reports to the master HATCPU sub-function • Reports the loss of any task on its CPU to the master HATCPU sub-function	All PACs	ALL

Task	Description	Card	CPU
	• Performs device initialization and control functions because of the CPU's hardware capabilities • Reports the loss of any task on its CPU to HATSYSTEM sub-function • Controls the LEDs on the SPC • Initializes and monitors the dedicated hardware on the SPCs	SPCs	0

Table 52. ST16 Resource Manager (RM) Subsystem

Task	Description	Card	CPU
Resource Manager Controller (RMCTRL)	Started by the SITPARENT task on system startup, and monitored by the HAT task for a failure, the RMCTRL performs the following functions at startup: • Initializes resources such as CPUs and memory • Requests updated card status from the CSP subsystem and updates the system card table • Communicates with all RMMGRs to request their most recent set of resource data	Active SPC	0
Resource Manager Managers (RMMGRs)	Started by the SITPARENT task, and monitored by the HAT tasks for failures, each RMMGR performs the following functions at startup: • Initializes the local resource data and local resource scratch space • Communicates with the SIT task on the local CPU to get its entire task table and the resources associated with each task • Gathers current resource utilization for each task • Sends the resource data to the RMCTRL task	All	All

Table 53. ST16 Virtual Private Networking (VPN) Subsystem

		Card		CPU	
Task	Description	w/o SR	w/ SR	w/o SR	w/ SR

		Card		CPU	
Task	**Description**	**w/o SR**	**w/ SR**	**w/o SR**	**w/ SR**
VPN Controller (VPNCTRL)	Created at system start-up, the VPN Controller: • Initiates the VPN Manager for each context • Informs the Session Controller task when there are additions or changes to contexts • Routes context-specific operation information to the appropriate VPN Manager • Performs VPN Manager recovery and saves all VPN related configuration information in the SCT task Only one Session Controller operates at any time.	Active SPC	Active SPC	0	0
VPN Manager (VPNMGR)	One VPN manager is started by the VPN Controller for each configured context (one is always present for the local context) and performs the following functions: • Performs IP address pool and subscriber IP address management • Performs all the context specific operations including but not limited to: UCM services, IP interfaces, the Address Resolution Protocol (ARP), IP address pool management, slow path forwarding, NPU flows, port Access Control Lists (ACLs), and logging • Provides IP interface address information for each context to the Session Controller	Active SPC (local context)Active PACs (cfg contexts)	Active SPC (local context)1st PAC (cfg contexts)	0 (all contexts)	0 (local context)All on 1st PAC (cfg contexts)

		Card		CPU	
Task	**Description**	**w/o SR**	**w/ SR**	**w/o SR**	**w/ SR**
Border Gateway Protocol (BGP)	The BGP task is created by the VPN Manager for each context that has enabled the BGP routing protocol (router bgp config-context CLI command). The BGP task is responsible for learning and redistributing routing information via the BGP protocol. This includes: • Maintaining the BGP peering connections • Applying any defined BGP routing policy	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.
Open Shortest Path First (OSPF)	The OSPF task is created by VPN Manager for each context that has enabled the OSPF routing protocol (router ospf config-context CLI command). The OSPF task is responsible for learning and redistributing routing information via the OSPF protocol. This includes: • Maintaining the OSPF neighboring relationship • LSA database maintenance • SPF calculations • Applying any defined OSPF routing policy	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.
Open Shortest Path First (OSPFv3)	The OSPFv3 task is created by VPN Manager for each context that has enabled the OSPF routing protocol (router ospfv3 config-context CLI command) The OSPFv3 task is responsible for learning and redistributing routing information via the OSPFv3 protocol. This includes: • Maintaining the OSPFv3 neighboring relationship • LSA database maintenance • OSPFv3 SPF calculations • Applying any defined OSPFv3 routing policy	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.

		Card		CPU	
Task	**Description**	**w/o SR**	**w/ SR**	**w/o SR**	**w/ SR**
Routing Information Protocol (RIP)	The RIP task is created by VPN Manager for each context that has enabled the RIP routing protocol (router rip config-context CLI command) The RIP task is responsible for learning and redistributing routing information via the RIP protocol. This includes: • maintaining the RIP database • sending periodic RIP update messages • applying any defined RIP routing policy	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.
ZEBOSTM OSPF Message	The Zebos task is created by VPN Manager for each context. The Zebos task maintains the routing table (RIB and FIB) for the context. This includes: • Static routing • Interfacing to the kernel for routing and interface updates • Redistributing routing information to dynamic routing protocols • Calculating nexthop reachability	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.

Table 54. ST16 Network Processing Unit (NPU) Subsystem

Task	Description	Card	CPU
NPU Controller (NPUCTRL)	Created at system start-up, the NPU Controller performs the following functions: • Monitors the state of NPU Managers in the system • Registers to receive notifications when NPU Manager crashes • Controls recovery operation • Provides a centralized location for CLI commands related to NPU Manager state Only one NPU Controller operates in the system at any time.	Active SPC	0

Task	Description	Card	CPU
NPU Manager (NPUMGR)	The NPU Manager task is created for every PAC installed and started. It performs the following functions: • Provides port configuration services to the CSP task • Provides interface binding and forwarding services to the VPN Manager • Provides flow insertion and removal services to Session Manager and AAA Manager tasks • Provides recovery services to the NPU Controller	SPCs/PACs	0

Table 55. ST16 Session Subsystem

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Session Controller (SESSCTRL)	Created at system start-up, the Session Controller task performs the following functions: • Acts as the primary point of contact for the Session Subsystem. Since it is aware of the other subsystems running within the system, the Session Controller acts as a proxy for the other components, or tasks, that make up the subsystem • Starts, configures, and coordinates the efforts of the Session Processing Subsystem sub-managers • Works with Resource Manager to start new Session Managers when all existing Session Managers exceed their capacity • Receives context information from VPN Managers • Distributes IP interface address information to other Session Processing Subsystem sub-managers • Manages Enhanced Charging Service, Content Filtering and URL Blacklisting services Only one Session Controller operates in the system at any time.	Active SPC	Active SPC	0	0

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Session Manager (SESSMGR)	Created by the Session Controller, the Session Manager performs the following functions: • Provides a subscriber processing system that supports multiple session types • Multiple Session Managers can run on a single CPU and/or can be distributed throughout any CPU present in the system • A single Session Manager can service sessions from multiple A11 Managers, ASNGW Manager, ASN PC Manager, GTPC Message Managers, and from multiple contexts • Protocol processing for A10/A11, GRE, R3, R4, R6, GTPU/GTPC, PPP, and Mobile IP • Manages Enhanced Charging Service, Content Filtering and URL Blacklisting services Session Managers are paired with AAA Managers.	Active PACs	All PACs except 1st	All except 0	All

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
A11 Manager (A11MGR)	Created by the Session Controller for each context in which a PDSN service is configured, the A11 Manager task performs the following functions: • Receives the R-P sessions from the PCF and distributes them to different Session Manager tasks for load balancing • Maintains a list of current Session Manager tasks to aid in system recovery The A11 Manager task is also known as the Signaling De-multiplexing task (SDT). ***Important:*** For ST16s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PAC.	Active PACs	1st PAC	0	Any (see **NOTE**)
ASN Gateway Manager (ASNGWMGR)	Created by the Session Controller, the ASN Gateway Manager performs the following functions: • Provides a subscriber processing system that supports multiple session types • Multiple ASNGW Managers can run on a single CPU and/or can be distributed throughout any CPU present in the system • A single ASN GW Manager can service sessions from multiple ASN PC Managers and multiple contexts • Protocol processing for R3, R4, R6, GRE tunneling, and Mobile IP	Active PACs		0	

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
ASN PC Manager (ASNPCMGR)	Created by the Session Controller, the ASN Paging Controller and Location Registry Manager performs the following functions: • Provides a subscriber processing system that supports multiple paging controller and location update session types • Multiple ASNPC Managers can run on a single CPU and/or can be distributed throughout any CPU present in the system • A single ASN GW Manager can service sessions from multiple contexts • Protocol processing for R3, R4, R6, GRE tunneling, and Mobile IP	Active PACs		0	
Authorization, Authentication, and Accounting (AAA) Manager (AAAMGR)	AAA Managers are paired with Session Managers (except the one running on the SPC) and perform the following functions: • Perform all AAA protocol operations and functions for subscribers and administrative users within the system • Act as a AAA client to AAA servers • Manage GTP Prime (GTPP) messaging with charging gateway functions (CGFs). • Multiple AAA Managers can run on a single CPU and/or can be distributed throughout any CPU present in the system. • AAA operations for the CLI are done through a AAA Manager running on the active SPC	Active PACs	All PACs except 1st	All except 0	All
		Active SPC (CLI only)	Active SPC (CLI only)	0	0
Charging Detail Record Module (CDRMOD)	CDRMOD is created by VPNCTRL proclet in v9.x releases and by the VPNMGR proclet in pre-v9.x releases. Receives EDR/UDR records from different ACSMGR instances in the system. Writes the received EDR/UDR records in files using the configured file naming conventions.	1st PAC	1st PAC	0	0

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Diameter GMB Application Manager (DGMBMGR)	DGMBMGR is created specifically for providing MBMS feature support for GGSN. It is instantiated when an MBMS policy CLI is configured in the GGSN Service configuration mode. DGMBMGR maintains the MBMS UE and bearer contexts. It handles the GMB interface over a Diameter connection to a BMSC Server for MBMS bearer sessions. DGMBMGR recovers by polling all SMGRs for MBMS session states and recreating the MBMS UE and MBMS bearer context information.	Active PACs		0	
Diameter Proxy (DIAMPROXY)	Diameter proxy is created by DIACTRL (which runs as part of VPNCTRIL). The number of DIAMPROXY tasks spawned is based on the configuration to use multiple or single proxies. If a single proxy is configured, only one DIAMPROXY task, which runs on demux PACs, is spawned for the entire chassis. If multiple proxies, are configured, one DIAMPROXY task is run per PAC. It performs the following functions: • Maintains diameter base connections to all peers configured in the system • Informs applications about any change in the connection status • Acts as a pass-through to the messages from application to the diameter server • Acts as a forwarding agent (doesn't maintain any queues) A single diameter proxy services multiple diameter applications.	Active PACs (see description)	Active PACs (see description)	All (see description)	All (see description)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Foreign Agent (FA) Manager (FAMGR)	Created by the Session Controller for each context in which an FA service is configured, the FA Manager performs the following functions: • Maintains a list of the FA-services available within the context and performs load-balancing for them • Performs load-balancing by routing incoming MIP calls between the FA Managers ***Important:*** For ST16s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PAC.	Active PACs	1st PAC	0	Any (see **NOTE**)
GPRS Tunneling Protocol Control (GTPC) Message Manager (GTPCMGR)	Created by the Session Controller for each context in which a GGSN service is configured, the GTPC Manager task performs the following functions: • Receives the GTP sessions from the SGSN and distributes them to different Session Manager tasks for load balancing • Maintains a list of current Session Manager tasks to aid in system recovery • Verifies validity of GTPC messages • Maintains a list of current GTPC sessions • Handles GTPC Echo messaging to/from SGSN	Active PACs		0	

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Home Agent (HA) Manager (HAMGR)	Created by the Session Controller for each context in which an HA service is configured, the HA Manager performs the following functions: • Receives Mobile IP sessions from the Foreign Agents (FAs) and distributes them to different Session Manager tasks • Maintains a list of current Session Manager tasks that aids in system recovery • Functions as the DemuxMgr and handles all the PMIP signaling packets • Functions as the Demuxmgr for MIPv6/MIPv4 HA ***Important:*** For ST16s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PAC.	Active PACs	1st PAC	0	Any (see **NOTE**)
IP Services Gateway Manager (IPSGMGR)	Created by the Session Controller, the IPSG Manager performs the following functions: • In Server mode, acts as a RADIUS server and supports Proxy functionality • In Snoop mode supports snooping RADIUS Accounting messages • Load balances requests among different SessMgrs • Activates and deactivates sessions	Active PACs		0	

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Layer 2 Tunneling Protocol Manager (L2TPMGR)	Created by the Session Controller for each context in which a LAC or LNS service is configured (additional Managers created as needed depending on loading), the L2TP Manager task performs the following functions: • L2TP processing • Maintaining protocol state machines for all L2TP sessions and tunnels • Triggering IPSec encryption for new L2TP tunnels as needed • Working with Session Managers to gracefully bring down tunnels ***Important:*** For ST16s with session recovery enabled, this demux manager is usually established on one of the CPUs on the first active PAC.	Active PACs	1st PAC	0	Any (see **NOTE**)
L2TP Demultiplexor Task (L2TPDEMUX)	Created by the Session Controller when a LNS service is created, only one L2TPDemux task is invoked for the entire system. This task performs the following functions: • De-multiplexes and forwards new incoming tunnel create requests to L2TPMgrs • Maintains information about current active tunnels in all L2TPMgrs • Load balances requests among L2TPMgrs ***Important:*** For ST16s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PAC.	Active PACs	1st PAC	0	Any (see **NOTE**)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Standard Routing Database (SRDB)	Eight SRDBs are created by the Session Controller when Content Filtering in the Enhanced Charging Service is enabled. A minimum of two PACs are required to initiate these eight tasks. SRDB performs the following functions: • The SRDB task receives the static database from the session controller. Each SRDB task loads two database volumes (one primary and one secondary) each. • The SRDB task is also responsible for storing the static DB.The SRDB task is used for rating the URL. It returns the proper category of the URL depending on the DB volumes and CSI (category set Index) stored on it. • The SRDB tasks are responsible for peer loading in case its peer fails. If both the SRDB task and it's peer fails, then loading is performed by the session controller.	Peer SRDBs evenly distributed across PACs			

Table 56. ST16 Platform Processes

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Card-Slot-Port Controller (CSPCTRL)	Manages physical chassis components.	SPC	SPC	0	0
Messenger Daemon (MSGD)	Implements the Name Service and related functions for the internal message passing system.	All	All	All	All
Name Service Controller (NSCONTROL)	As part of the Messenger process, NSControl provides a reliable channel for tasks to send control messages to the Messenger Daemon.	All	All	All	All
Daughter Card Controller (DCARDCTRL)	The daughter card controller is responsible for spawning daughter card managers during system initialization and monitoring daughter card managers during system steady state execution. The daughter card controller is responsible for spawning daughter card managers in the case of a daughter card manager task failure.	Active SPC	Active SPC	0	0

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Daughter Card Manager (DCARDMGR)	The daughter card manager is responsible for managing IPsec Security Associations for AH and ESP based sessions. The daughter card manager is also responsible for interfacing to the on-board hardware accelerated cryptographic chip which executes cryptographic algorithms associated with the given IPsec Security Associations.	All PACs	All PACs	0	0
Distributed Host Manager (DHMGR)	Started automatically on each CPU by SITPARENT. Coordinates establishment of locally terminated TCP, SCTP, and UDP connections on behalf of multi-instanced tasks such as Diameter endpoints among SESSMGR tasks.	All	All	All	All
IPSec Controller (IPSECCTRL)	The IPSec controller is started by SIT on system startup regardless of configuration and performs the following functions: • It starts IPSECMGR tasks based on configuration and maintains its list for task recovery. • It receives user configuration for IPSec and maintains the same. • It also manages the configured IPSec crypto maps and its assignment to ipsecmgrs. • It interfaces with the VPNMGR task for required IPSec configuration parameters such as IP Access-lists, IP pools, interface addresses, interface state notifications, etc.	Active SPC	Active SPC	0	0
IPSec Manager (IPSECMGR)	Created by the Session Controller, the IPSECMGR performs the following functions: • Establishes and manages secure IKEv1, IKEv2 and IPSec data tunnels.	PAC	PAC	All	All
Driver Controller (DRVCTRL)	The driver controller is responsible for the centralizing accesses to many of the system device drivers. It also performs temperature & voltage monitoring.	SPCs	SPCs	0	0
Switch Fabric Task (SFT)	The switch fabric task is responsible for monitoring the switch fabric and the gigabit Ethernet control plane.	PACs	PACs	0	0
Shared Configuration Task (SCT)	Performs the redundant storage of configuration information and other state information in an in-memory database.	SPCs	SPCs	0	0
Recovery Control Task (RCT)	Monitors tasks/managers/facilities across the system and performs recovery in the event of a failure.	SPCs	SPCs	0	0
Utilities Configuration Manager (UCM)	DHCPD, DNS, FTPD, INETD, NTPD, PING, RLOGIN, SFTPD, SFTP-SERVER, SNMPD, SSH, SSHD, TELNET, TELNETD, TFTPD, TRACEROUTE	Active SPC	Active SPC	0	0

Table 57. ST16 Management Processes

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Bulk Statistic Manager (BULKSTAT)	Performs a periodic statistic polling/gathering function ("bulk statistics") and handles the transfer of this data to external management systems.	Active SPC	Active SPC	0	0
Event Log Daemon (EVLOGD)	Handles event logging functions including the interface to external syslogd servers and the internal event logs.	Active SPC	Active SPC	0	0
ORB Service (ORBS)	The orbs task performs the following functions: • Application Servers (EMS) requests orbs to perform Element Management Functions on the ST16 using secure IIOP. orbs then interacts with concerned Controller Tasks to execute the function. • The response/errors from the execution is interpreted, formulated into EMF response and handed over to Application Server (EMS). The orbs task is also known as the ORB Element Manager (ORBEM).	Active SPC	Active SPC	0	0
ORB Notification Service (ORBNS)	The orbns task performs the following functions: There are different types of event that can continually occur within Boxer. The Application Servers (EMS) may be interested in being notified of occurrences of these events. Such Application Servers (EMS) may register with orbs (ORBEM) subscribing to types of event they are interested in. As the events occur, the concerned Controller Task would notify orbs (ORBEM), which in-turn will notify the subscribing Application Servers (EMS).	Active SPC	Active SPC	0	0
Simple Network Management Protocol (SNMP)	Handles inboard SNMP operations if configured and sends SNMP notifications (traps) if enabled.	Active SPC	Active SPC	0	0
Threshold Server (THRESHOLD)	Handles monitoring of "threshold crossing alerts", if configured. Polls the needed statistics/variables, maintains state, and generates log messages/SNMP notification of threshold crossings.	Active SPC	Active SPC	0	0

ASR 5000 Subsystems

The following tables describe managers and tasks performing within the specified subsystems on an ASR 5000.

Important: Variations regarding how the managers and tasks are distributed based on session recovery are included in the Card and CPU columns in some tables. Tables without these indicators are applicable to ASR 5000s

with and without session recovery. The ASR 5000 dynamically distributes processes, tasks, and managers on startup. The following tables list the typical locations but variations can occur depending on available resources.

Table 58. ASR 5000 System Initiation Subsystem

Task	Description	Card	CPU
SITMAIN	Initiated at system start-up, the SITMAIN task performs the following functions: • Reads and provides startup configuration to other SIT components • Starts SITREAP sub-function • Maintains CPU state information	All	All
SITPARENT sub-function	• Starts SPCs/SMCs in either active or standby mode • Registers tasks with HAT task • Notifies CSP task of CPU startup completion • Brings PACs up in standby mode **Important:** SITPARENT replaces the sub-functions SITPAC, SITSPC and SITTAC.	All	All
SITREAP sub-function	Shuts down tasks as required	All	All

Table 59. ASR 5000 High Availability Subsystem

Task	Description	Card	CPU
HAT System Controller (HATSYSTEM)	This is the main HAT task that will control all the HAT sub-function tasks in the system. It is initiated on system start-up and performs the following functions: • Initializes system components (such as the Gigabit Ethernet switches and switch fabric) • Monitors system components such as fans for state changes • Triggers actions for redundancy in the event of fault detection The HAT subsystem on the redundant SPC/SMC mirrors the HAT subsystem on the active SPC/SMC.	SMCs	0

Task	Description	Card	CPU
HATCPU	• Performs device initialization and control functions because of the CPUs hardware capabilities • Reports the loss of any task on its CPU to HATSYSTEM sub-function • Controls the LEDs on the PAC • Initializes and monitors the dedicated hardware on PACs	All PSCs	0
	• Collects CPU monitoring information periodically and reports to the master HATCPU sub-function • Reports the loss of any task on its CPU to the master HATCPU sub-function	All PSCs	ALL
	• Performs device initialization and control functions because of the CPU's hardware capabilities • Reports the loss of any task on its CPU to HATSYSTEM sub-function • Controls the LEDs on the SMC • Initializes and monitors the dedicated hardware on the SMCs	SMCs	0

Table 60. ASR 5000 Resource Manager (RM) Subsystem

Task	Description	Card	CPU
Resource Manager Controller (RMCTRL)	Started by the SITPARENT task on system startup, and monitored by the HAT task for a failure, the RMCTRL performs the following functions at startup: • Initializes resources such as CPUs and memory • Requests updated card status from the CSP subsystem and updates the system card table • Communicates with all RMMGRs to request their most recent set of resource data	Active SMC	0
Resource Manager Managers (RMMGRs)	Started by the SITPARENT task, and monitored by the HAT tasks for failures, each RMMGR performs the following functions at startup: • Initializes the local resource data and local resource scratch space • Communicates with the SIT task on the local CPU to get its entire task table and the resources associated with each task • Gathers current resource utilization for each task • Sends the resource data to the RMCTRL task	All	All

Table 61. ASR 5000 Virtual Private Networking (VPN) Subsystem

		Card		CPU	
Task	**Description**	**w/o SR**	**w/ SR**	**w/o SR**	**w/ SR**
VPN Controller (VPNCTRL)	Created at system start-up, the VPN Controller: • initiates the VPN Manager for each context • informs the Session Controller task when there are additions or changes to contexts • routes context specific operation information to the appropriate VPN Manager • performs VPN Manager recovery and saves all VPN related configuration information in the SCT task Only one Session Controller operates at any time	Active SMC	Active SMC	0	0
VPN Manager (VPNMGR)	One VPN manager is started by the VPN Controller for each configured context (one is always present for the local context) and performs the following functions: • Performs IP address pool and subscriber IP address management • Performs all the context specific operations including but not limited to: UCM services, IP interfaces, the Address Resolution Protocol (ARP), IP address pool management, slow path forwarding, NPU flows, port Access Control Lists (ACLs), and logging • Provides IP interface address information for each context to the Session Controller	Active SMC (local context)Active PSCs (cfg contexts)	Active SMC (local context)1st PSC (cfg contexts)	0 (all contexts)	0 (local context)All on 1st PSC (cfg contexts)

		Card		CPU	
Task	**Description**	**w/o SR**	**w/ SR**	**w/o SR**	**w/ SR**
Border Gateway Protocol (BGP)	The BGP task is created by the VPN Manager for each context that has enabled the BGP routing protocol (“router bgp” config-context CLI command). The BGP task is responsible for learning and redistributing routing information via the BGP protocol. This includes: • maintaining the BGP peering connections • applying any defined BGP routing policy	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.
Open Shortest Path First (OSPF)	The OSPF task is created by VPN Manager for each context that has enabled the OSPF routing protocol (“router ospf” config-context CLI command). The OSPF task is responsible for learning and redistributing routing information via the OSPF protocol. This includes: • maintaining the OSPF neighboring relationship • LSA database maintenance • SPF calculations • applying any defined OSPF routing policy	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.
Open Shortest Path First (OSPFv3)	The OSPFv3 task is created by VPN Manager for each context that has enabled the OSPF routing protocol (“router ospfv3” config-context CLI command) The OSPFv3 task is responsible for learning and redistributing routing information via the OSPFv3 protocol. This includes: • maintaining the OSPFv3 neighboring relationship • LSA database maintenance • OSPFv3 SPF calculations • applying any defined OSPFv3 routing policy	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.

		Card		CPU	
Task	**Description**	**w/o SR**	**w/ SR**	**w/o SR**	**w/ SR**
Routing Information Protocol (RIP)	The RIP task is created by VPN Manager for each context that has enabled the RIP routing protocol (“router rip” config-context CLI command) The RIP task is responsible for learning and redistributing routing information via the RIP protocol. This includes: • maintaining the RIP database • sending periodic RIP update messages • applying any defined RIP routing policy	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.
ZEBOSTM OSPF Message	The Zebos task is created by VPN Manager for each context. The Zebos task is responsible for maintaining the routing table for the context. This includes: • maintaining the routing table (RIB and FIB) • static routing • interfacing to the kernel for routing & interface updates • redistributing routing information to dynamic routing protocols • calculating nexthop reachability	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.

Table 62. ASR 5000 Network Processing Unit (NPU) Subsystem

Task	Description	Card	CPU
NPU Controller (NPUCTRL)	Created at system start-up, the NPU Controller performs the following functions: • Monitors the state of NPU Managers in the system • Registers to receive notifications when NPU Manager crashes • Controls recovery operation • Provides a centralized location for CLI commands related to NPU Manager state Only one NPU Controller operates in the system at any time.	Active SMC	0

Task	Description	Card	CPU
NPU Manager (NPUMGR)	The NPU Manager task is created for every PSC installed and started and it performs the following functions: • Provides port configuration services to the CSP task • Provides interface binding and forwarding services to the VPN Manager • Provides flow insertion and removal services to Session Manager and AAA Manager tasks • Provides recovery services to the NPU Controller	SMCs/PSCs	0

Table 63. ASR 5000 Session Subsystem

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Session Controller (SESSCTRL)	Created at system start-up, the Session Controller task performs the following functions: • Acts as the primary point of contact for the Session Subsystem. Since it is aware of the other subsystems running within the system, the Session Controller acts as a proxy for the other components, or tasks, that make up the subsystem • Starting, configuring, and coordinating the efforts of the Session Processing Subsystem sub-managers • Works with Resource Manager to start new Session Managers when all existing Session Managers exceed their capacity • Receives context information from VPN Managers • Distributes IP interface address information to other Session Processing Subsystem sub-managers • Manages Enhanced Charging Service, Content Filtering and URL Blacklisting services Only one Session Controller operating in the system at any time.	Active SMC	Active SMC	0	0

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Session Manager (SESSMGR)	Created by the Session Controller, the Session Manager performs the following functions: • Provides a subscriber processing system that supports multiple session types • Multiple Session Managers can run on a single CPU and/or can be distributed throughout any CPU present in the system • A single Session Manager can service sessions from multiple A11 Managers, ASNGW Manager, ASN PC Manager, GTPC Message Managers, and from multiple contexts • Protocol processing for A10/A11, GRE, R3, R4, R6, GTPU/GTPC, PPP, and Mobile IP • Manages Enhanced Charging Service, Content Filtering and URL Blacklisting services Session Managers are paired with AAA Managers.	All PSCs	All PSCs except 1st	0	0 on all PSCs except 1st

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
A11 Manager (A11MGR)	Created by the Session Controller for each context in which a PDSN service is configured, the A11 Manager task performs the following functions: • Receives the R-P sessions from the PCF and distributes them to different Session Manager tasks for load balancing • Maintains a list of current Session Manager tasks to aid in system recovery The A11 Manager task is also known as the Signaling De-multiplexing task (SDT). ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PSC.	Active PSCs	1st PSC	0	Any (see **NOTE**)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Access Link Control Application Part Manager (ALCAPMgr)	The ALCAP Mgr tasks starts when an ALCAP service configuration is detected. There can be multiple instances of this task for load sharing. All ALCAP Managers will have all the Active ALCAP Services configured in HNB-GW service and will be identical in configuration and capabilities. • It runs the ALCAP protocol stack and handles the IuCS-over-ATM associations • Maintains AAL2 node entity databases • Provides nodal functions for IuCS-over-ATM interface on ALCAP protocol ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PSC. The HNBMGRs should not be started on a PSC which has the HNB DEMUX MGR started.	Active PSC	Active PSC (see **NOTE**)	0	Any (see **NOTE**)
ASN Gateway Manager (ASNGWMGR)	Created by the Session Controller, the ASN Gateway Manager performs the following functions: • Provides a subscriber processing system that supports multiple session types • Multiple ASNGW Managers can run on a single CPU and/or can be distributed throughout any CPU present in the system • A single ASN GW Manager can service sessions from multiple ASN PC Managers and multiple contexts • Protocol processing for R3, R4, R6, GRE tunneling, and Mobile IP	Active PSCs		0	

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
ASN PC Manager (ASNPCMGR)	Created by the Session Controller, the ASN Paging Controller and Location Registry Manager performs the following functions: • Provides a subscriber processing system that supports multiple paging controller and location update session types • Multiple ASNPC Managers can run on a single CPU and/or can be distributed throughout any CPU present in the system • A single ASN GW Manager can service sessions from multiple contexts • Protocol processing for R3, R4, R6, GRE tunneling, and Mobile IP	Active PSCs		0	
Authorization, Authentication, and Accounting (AAA) Manager (AAAMGR)	AAA Managers are paired with Session Managers (except the one running on the SMC) and perform the following functions: • Performs all AAA protocol operations and functions for subscribers and administrative users within the system • Acts as a AAA client to AAA servers • Manages GTP Prime (GTPP) messaging with charging gateway functions (CGFs). • Multiple AAA Managers can run on a single CPU and/or can be distributed throughout any CPU present in the system. • AAA operations for the CLI are done through a AAA Manager running on the active SMC	Active PSCs	All PSCs except 1st	All except 0	All
		Active SMC (CLI only)	Active SMC (CLI only)	0	0

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Charging Detail Record Module (CDRMOD)	CDRMOD is created by VPNCTRL proclet in v90+ releases and by the VPNMGR proclet in pre-v90 releases. Responsible for receiving EDR/UDR records from different ACSMGR instances in the system. Responsible for writing the received EDR/UDR records in files using the configured file naming conventions.	1st PSC	1st PSC	0	0
Diameter GMB Application Manager (DGMBMGR)	DGMBMGR is created specifically for providing MBMS feature support for GGSN. It is instantiated when an MBMS policy CLI is configured in the GGSN Service configuration mode. DGMBMGR maintains the MBMS UE and bearer contexts. It handles the GMB interface over a Diameter connection to a BMSC Server for MBMS bearer sessions. DGMBMGR recovers by polling all SMGRs for MBMS session states and recreating the MBMS UE and MBMS bearer context information.	Active PSCs		0	
Diameter Proxy (DIAMPROXY)	Diameter proxy is created by DIACTRL (which runs as part of VPNCTRIL) and the number of DIAMPROXY tasks spawned is based on the configuration to use “multiple” or “single” proxies. In instances that a single proxy is configured, only one DIAMPROXY task is spawned for the entire chassis and runs on demux PACs. When multiple proxies are configured, one DIAMPROXY task is run per PAC. It performs the following functions: • Maintains diameter base connections to all peers configured in the system • Informs applications about any change in the connection status • Acts as a pass-through to the messages from application to the diameter server • Just acts as a forwarding agent (doesn't maintain any queues) A single diameter proxy is used to service multiple diameter applications	Active PSCs (see description)	Active PSCs (see description)	All (see description)	All (see description)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
eGTP Egress Manager	Created by Session Controller for each context in which an egtp-service of interface type sgw-egress or MME is configured. The egtpinmgr performs the following functions: • Handles certain EGTP messages from SGW, PGW • Maintains list of current EGTP sessions • Maintains list of current Session Manager tasks which aids in session recovery • Handles GTP Echo messaging ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PSC.	PSC	1st PSC	All	Any (see **NOTE**)
eGTP Ingress Manager	Created by Session Controller for each context in which an egtp-service of interface type sgw-ingress or pgw-ingress is configured. The egtpinmgr performs the following functions: • Receives EGTP sessions from MME/S4 SGSN/SGW and distributes them to different Session Manager tasks for load balancing • Maintains list of current EGTP sessions • Maintains list of current Session Manager tasks which aids in session recovery • Handles GTP Echo messaging ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PSC.	PSC	1st PSC	All	Any (see **NOTE**)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Foreign Agent (FA) Manager (FAMGR)	Created by the Session Controller for each context in which an FA service is configured, the FA Manager performs the following functions: • Maintains a list of the FA-services available within the context and performs load-balancing for them • Performs load-balancing by routing incoming MIP calls between the FA Managers ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PSC.	Active PSCs	1st PSC	0	Any (see **NOTE**)
GPRS Tunneling Protocol Control (GTPC) Message Manager (GTPCMGR)	Created by the Session Controller for each context in which a GGSN service is configured, the GTPC Manager task performs the following functions: • Receives the GTP sessions from the SGSN and distributes them to different Session Manager tasks for load balancing • Maintains a list of current Session Manager tasks to aid in system recovery • Verifies validity of GTPC messages • Maintains a list of current GTPC sessions • Handles GTPC Echo messaging to/from SGSN'	Active PSCs		0	

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
GTP-U Manager (GTPUMGR)	Created by the Session Controller for each context in which a GTPU service is configured, the GTPU Manager performs the following functions: • Maintains a list of the GTPU-services available within the context and performs load-balancing (of only Error-Ind) for them • Support for GTPU Echo handling • Path Failure detection on no response for GTPU echo • Error-Ind reception and demuxing it to a particular SMGR • Default GTPU listener. GTPUMGR will process GTPU packets with invalid TEID Above features are supported for both GTPUv0 and GTPUv1. ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PSC.	Active PSCs	1st PSC	0	Any (see **NOTE**)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
HNB Demux Manager (HNBDemux)	The HNB Demux Manager is started as part of HNB-GW service creation procedure. There will be only one HNBDEMUX MGR in the chassis. • Distributes incoming Iuh connections to HNB Mgrs in the system • Aware of all the active HNB-GW services in the system ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PSC but should not be created on the same PSC that has HNB Manager.	PSC	PSC (see **NOTE**)	0	Any (see **NOTE**)
HNB Manager (HNBMgr)	The HNB Mgr tasks starts when an HNB-GW service configuration is detected. There can be multiple instances of this task for load sharing. All HNB Managers will have all the Active HNB-GW Services configured and will be identical in configuration and capabilities. • It runs the SCTP protocol stack • handles the SCTP associations • Maintains Home-NodeB databases • Provides nodal functions for Iuh interface on SCTP protocol ***Important:*** For ASR 5000s with session recovery enabled, this manager is usually established on one of the CPUs on the 1st active PSC. The HNBMGRs should not be started on a PSC which has the HNB DEMUX MGR started.	Active PSC	Active PSC (see **NOTE**)	0	Any (see **NOTE**)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Home Agent (HA) Manager (HAMGR)	Created by the Session Controller for each context in which an HA service is configured, the HA Manager performs the following functions: • Receives Mobile IP sessions from the Foreign Agents (FAs) and distributes them to different Session Manager tasks • Maintains a list of current Session Manager tasks that aids in system recovery • Functions as the DemuxMgr – handles all the PMIP signaling packets. • HAMgr also functions as the Demuxmgr for MIPv6/MIPv4 HA. ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PSC.	Active PSCs	1st PSC	0	Any (see **NOTE**)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
IMSI Manager for MME (IMSIMgr)	The IMSI Mgr tasks starts when an MME service configuration is detected. There will be only one instance of this taks.IMSI Manager has the following functions:Signaling De-multiplexer: Selects which SessMgr to use for new subscriber sessions.IMSI-to-SessMgr resolutionMaintains and reports MME related Demux statistics on events like Attach by IMSI, Attach by GUTI etc. IMSIMgr can interact with following different tasks in the system: • Session Controller • MME Manager • Session Manager ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PSC. The IMSIMgr will not start on a PSC in which SessMgrs are started.	Active PSC	Active PSC (see **NOTE**)	0	Any (see **NOTE**)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
International Mobile Subscriber Identity Manager for SGSN (IMSIMgr)	Started by the Session Controller, the IMSIMgr performs the following functions • Selects SessMgr, when not done by LinkMgr or SGTPCMgr, for calls sessions based on IMSI/P-TMSI. • Load-balances across SessMgrs to select one for assigning subscriber sessions to. • Maintains records for all subscribers on the system. • Maintains mapping between the IMSI/P-TMSI and SessMgrs. ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active deumux PSC. The IMSIMgr will not start on a PSC in which SessMgrs are already started.	Active PSC	Active PSC (see **NOTE**)	0	Any (see **NOTE**)
IP Services Gateway Manager (IPSGMGR)	Created by the Session Controller, the IPSG Manager performs the following functions: • In Server mode, acts as a RADIUS server, and supports Proxy functionality • In Snoop mode supports snooping RADIUS Accounting messages • Load balances requests among different SessMgrs • Activates and deactivates sessions	Active PSCs		0	

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Layer 2 Tunneling Protocol Manager (L2TPMGR)	Created by the Session Controller for each context in which a LAC or LNS service is configured, (additional Managers created as needed depending on loading) the L2TP Manager task performs the following functions: • Responsible for all aspects of L2TP processing • Maintain protocol state machines for all L2TP sessions and tunnels • Trigger IPSec encryption for new L2TP tunnels as needed • Work with Session Managers to gracefully bring down tunnels ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PSC.	Active PSCs	1st PSC	0	Any (see **NOTE**)
L2TP Demultiplexor Task (L2TPDEMUX)	Created by the Session Controller when a LNS service is created, only one L2TPDemux task is invoked for the entire system. This task performs the following functions: • De-multiplexes and forwards new incoming tunnel create requests to L2TPMgrs • Maintains information about current active tunnels in all L2TPMgrs • Load balances requests among L2TPMgrs ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PSC.	Active PSCs	1st PSC	0	Any (see **NOTE**)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Mobile Access Gateway Manager	Created by the Session Controller when the first MAG service is created in a context. This task performs the following functions: • Sends and receives PMIP control messages (PBU/PBA) • Adds an NPU flow to receive MIPv6 PBA packets. This flow is identical to the flow used in the HAMgr • Maintains the Binding Update List used to keep track of the mobile node's bindings: • MN-ID • APN • Home Network Prefix and prefix length • IPv6 LMA address • IPv4 Home Address • Originates PBU based on trigger received from the Session Manager during error conditions • Receives PBA and forwards it to Session Manager • Debugging facility – "magmgr" and "mobile-ipv6"	PSC		Same as VPN Mgr.	

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Mobility Management Entity Demux Manager (MMEDemux)	The MME Demux Manager is started as part of MME service creation procedure. There will be only one MME DEMUX MGR in the chassis. • Distributes incoming S1-MME SCTP connections to MME Mgrs in the system • Aware of all the active MME services in the system ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PSC but should not be created on the same PSC that has MME Manager.	PSC	PSC (see **NOTE**)	0	Any (see **NOTE**)
Mobility Management Entity Manager (MMEMgr)	The MME Mgr tasks starts when an MME service configuration is detected. There can be multiple instances of this task for load sharing. All MME Managers will have all the Active MME Services configured and will be identical in configuration and capabilities. • It runs the SCTP protocol stack • Handles the SCTP associations • Maintains TA List • Manage eNodeB databases • Provides nodal functions for S1-MME protocol ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active PSC. The MMEMGRs should not be started on a PSC which has the MME DEMUX MGR started.	Active PSC	Active PSC (see **NOTE**)	0	Any (see **NOTE**)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
SGSN GPRS Tunneling Protocol Control message Manager (SGTPCMgr)	Created by the Session Controller for each VPN context in which an SGSN service is configured, the SGTPC Manager task performs the following functions: • Terminates Gn/Gp and GTP-U interfaces from peer GGSNs and SGSNs for SGSN Services. • Terminates GTP-U interfaces from RNCs for IuPS Services. • Controls standard ports for GTP-C and GTP-U. • Processes and distributes GTP-traffic received from peers on these ports. • Performs all node level procedures associated with Gn/Gp interface. ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active demux PSC. The IMSIMgr will not start on a PSC in which SessMgrs are already started.	Active PSC	Active Demux PSC (see **NOTE**)	0	Any (see **NOTE**)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
SGSN Master Manager (MMgr)	MMgr is created upon provisioning of SS7RDs/SCCP-NWs/etc., The Session Controller provides the initial system configuration which includes a detailed description of each distributed protocol layer, its resources sets, and a list of its service user protocol layers and service provider protocol layers. The MMgr task runs in duplex mode (active/standby) to perform the following functions: • Single instanced • Handles nodal SS7, Iu, and Gb functionality. • Implements master LinkMgr functionality for SS7 route status aggregation. • Implements master LinkMgr functionality for RNC and BSC status aggregation. ***Important:*** For ASR 5000s with session recovery enabled, this demux manager is usually established on one of the CPUs on the 1st active demux PSC. The IMSIMgr will not start on a PSC in which SessMgrs are already started.	Active PSC	Active Demux PSC (see **NOTE**)	0	Any (see **NOTE**)
SS7 Link Manager (LinkMgr)	Created by the Session Controller when the first SS7RD (routing domain) is activated, the LinkMgr performs the following functions: • Multi-instanced for redundancy and scaling purposes. • Provides SS7 and Gb connectivity to the platform. • Routes per subscriber signalling across the SS7 (including Iu) and Gb interfaces to the SessMgr.	Any Active PSC Not Running an MMgr	Any Active non- demux PSC		

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Standard Routing Database (SRDB)	Eight SRDBs are created by the Session Controller when Content Filtering in the Enhanced Charging Service is enabled. A minimum of two PSCs or PSC2s are required to initiate these eight tasks. SRDB performs the following functions: • The SRDB task receives the static database from the session controller. Each SRDB task loads two database volumes (one primary and one secondary). The SRDB task also stores the static DB. • The SRDB task rates the URL. It returns the proper category of the URL depending on the DB volumes and CSI (category set Index) stored on it. • The SRDB tasks perform peer loading in case its peer fails. If both the SRDB task and its peer fail, the session controller performs the loading.	Peer SRDBs evenly distributed across PSCs			

Table 64. ASR 5000 Platform Processes

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Card-Slot-Port Controller (CSPCTRL)	Manages physical chassis components.	SMC	SMC	0	0
Messenger Daemon (MSGD)	Implements the Name Service and related functions for the internal message passing system.	All	All	All	All
Name Service Controller (NSCONTROL)	As part of the Messenger process, NSControl provides a reliable channel for tasks to send control messages to the Messenger Daemon.	All	All	All	All
Daughter Card Controller (DCARDCTRL)	The daughter card controller spawns daughter card managers during system initialization and monitors daughter card managers during system steady state execution. The daughter card controller spawns daughter card managers in case a daughter card manager task fails.	Active SMC	Active SMC	0	0

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Daughter Card Manager (DCARDMGR)	The daughter card manager is responsible for managing IPsec Security Associations for AH and ESP based sessions. The daughter card manager is also responsible for interfacing to the on-board hardware accelerated cryptographic chip which executes cryptographic algorithms associated with the given IPsec Security Associations.	All PSCs	All PSCs	0	0
Distributed Host Manager (DHMGR)	Started automatically on each CPU by SITPARENT. Coordinates establishment of locally terminated TCP, SCTP, and UDP connections on behalf of multi-instanced tasks such as Diameter endpoints among SESSMGR tasks.	All	All	All	All
Driver Controller (DRVCTRL)	The driver controller centralizes accesses to many of the system device drivers. It also performs temperature and voltage monitoring.	SMCs	SMCs	0	0
Hard Drive Controller (HDCTRL)	The hard drive controller controls/manages the RAID 1 array spanning the SMCs.	SMCs	SMCs	0	0
IPSec Controller (IPSECCTRL)	The IPSec controller is started by SIT on system startup regardless of configuration and performs the following functions: • Starts IPSECMGR tasks based on configuration and maintains its list for task recovery. • Receives and maintains user configuration for IPSec. • Manages the configured IPSec crypto maps and its assignment to IPSECMGRs. • Interfaces with the VPNMGR task for required IPSec configuration parameters such as IP Access-lists, IP pools, interface addresses, and interface state notifications.	Active SMC	Active SMC	0	0
IPSec Manager (IPSECMGR)	Created by the Session Controller, the IPSECMGR establishes and manages secure IKEv1, IKEv2 and IPSec data tunnels.	PSC	PSC	All	All
Recovery Control Task (RCT)	Monitors tasks/managers/facilities across the system and performs recovery in the event of a failure.	SMCs	SMCs	0	0
Shared Configuration Task (SCT)	Performs the redundant storage of configuration information and other state information in an in-memory database.	SMCs	SMCs	0	0
Switch Fabric Task (SFT)	The switch fabric task monitors the switch fabric and the gigabit Ethernet control plane.	PSCs	PSCs	0	0
Utilities Configuration Manager (UCM)	DHCPD, DNS, FTPD, INETD, NTPD, PING, RLOGIN, SFTPD, SFTP-SERVER, SNMPD, SSH, SSHD, TELNET, TELNETD, TFTPD, TRACEROUTE	Active SMC	Active SMC	0	0

Table 65. ASR 5000 Management Processes

Task	Description	Card	CPU

		w/o SR	w/ SR	w/o SR	w/ SR
Bulk Statistic Manager (BULKSTAT)	Performs a periodic statistic polling/gathering function (bulk statistics) and handles the transfer of this data to external management systems.	Active SMC	Active SMC	0	0
Event Log Daemon (EVLOGD)	Handles event logging functions including the interface to external syslogd servers and the internal event logs.	Active SMC	Active SMC	0	0
ORB Service (ORBS)	The ORBS task is also known as the ORB Element Manager (ORBEM). Application Servers (EMS) request ORBS to perform Element Management Functions on the system using secure IIOP. ORBS then interacts with concerned Controller Tasks to execute the function. The response/errors from the execution is interpreted, formulated into EMF response, and handed over to Application Server (EMS).	Active SMC	Active SMC	0	0
ORB Notification Service (ORBNS)	The ORBNS task performs the following functions: • There are different types of event that can continually occur within Boxer. • The Application Servers (EMS) may be notified of occurrences of these events. • Such Application Servers (EMS) may register with ORBS (ORBEM) subscribing to the types of events they are interested in. • As the events occur, the concerned Controller Task notifies ORBS (ORBEM), which then notifies the subscribing Application Servers (EMS).	Active SMC	Active SMC	0	0
Session Trace Collection Task (SESSTRC)	The session trace task implements the standards-based session trace functionality. The session trace task manages both CLI and signaling-based subscriber traces. It collects messages to be traced and generates trace files as needed. It uploads trace files to the Trace Collection Entity as needed.	Active SMC	Active SMC	0	0
Simple Network Management Protocol (SNMP)	Handles inboard SNMP operations if configured, and sends SNMP notifications (traps) if enabled.	Active SMC	Active SMC	0	0
Threshold Server (THRESHOLD)	Handles monitoring of threshold crossing alerts, if configured. Polls the needed statistics/variables, maintains state, and generates log messages/SNMP notification of threshold crossings.	Active SMC	Active SMC	0	0