



Cisco ASR 5000 System Administration Guide

Version 12.2

Last Updated April 30, 2012

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-25553-02

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to correct the interference at their own expense.

Modifications to this product not authorized by Cisco could void the FCC approval and negate your authority to operate the product.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco ASR 5000 System Administration Guide

© 2012 Cisco Systems, Inc. and/or its affiliated entities. All rights reserved.

CONTENTS

About this Guide	xiii
Conventions Used	xiv
Supported Documents and Resources	xvi
Related Documentation	xvi
Obtaining Documentation	xvi
Contacting Customer Support	xvii
Understanding System Operation and Configuration	19
Terminology	20
Contexts	20
Ports	20
Logical Interface	20
Management Interface	20
Bindings	21
Services	21
AAA Servers	22
Subscribers	22
How the System Selects Contexts	24
Context Selection for Context-level Administrative User Sessions	24
Context Selection for Subscriber Sessions	27
Understanding the ASR 5000 Boot Process	28
Understanding Configuration Files	30
IP Address Notation	32
IPv4 Dotted-Decimal Notation	32
IPv6 Colon-Separated Notation	32
CIDR Notation	32
Alphanumeric Strings	34
Character Set	34
Quoted Strings	35
Getting Started	37
ASR 5000 Configuration	38
Using the ASR 5000 Quick Setup Wizard	39
Using the CLI for Initial Configuration	44
Configuring the System for Remote Access	48
Configuring the Management Interface with a Second IP Address	51
Configuring System Settings	53
Configuring a Second Management Interface	54
Verifying and Saving Your Interface and Port Configuration	55
Configuring System Timing	56
Setting the System Clock and Time Zone	56
Verifying and Saving Your Clock and Time Zone Configuration	56
Configuring Network Time Protocol Support	57
Verifying the NTP Configuration	57
Configuring Transmit Timing Source	59
Configure BITS as the Timing Source	59
Configure Line-timing as the Timing Source	60

Configure Both BITS and Line as Timing Sources.....	60
Confirming the Timing Source	61
Enabling CLI Timestamping.....	61
Configuring System Administrative Users	61
Configuring Context-level Administrative Users	62
Configuring Context-level Security Administrators.....	62
Configuring Context-level Administrators.....	62
Configuring Context-level Operators.....	63
Configuring Context-level Inspectors	63
Verifying Context-level Administrative User Configuration	63
Configuring Local-User Administrative Users.....	64
Verifying Local-User Configuration.....	64
Configuring TACACS+ for System Administrative Users	65
Operation	65
User Account Requirements	65
TACACS+ User Account Requirements	65
User Account Requirements	66
Configuring TACACS+ AAA Services	67
Verifying the TACACS+ Configuration	68
Configuring a Chassis Key	69
Configuring Virtual MAC Addresses	70
Verifying Virtual MAC Address Configuration.....	71
Configuring Packet Processing and Line Card Availability.....	72
Verifying Packet Processing and Line Card Configurations.....	74
Configuring Line Card and SPIO Port Redundancy	75
Enabling Line Card and SPIO Port Redundancy	77
Verifying Line Card and SPIO Port Redundancy	78
Configuring Line Card and SPIO Port Redundancy Auto-Recovery.....	78
Verifying Line Card and SPIO Port Redundancy Auto-Recovery	79
Configuring ASR 5000 Link Aggregation.....	81
LAG and Master Port.....	81
LAG and Port Redundancy	81
LAG and Multiple Switches	81
Multiple Switches without L2 Redundancy.....	82
Multiple Switches with L2 Redundancy or Active-Active Mode	82
Port States for Auto-Switch	83
Hold Time	83
Preference and Revertive Mode	84
Auto-Switch Criteria	84
Distribution Options.....	84
QGLC Link Aggregation	84
Requirements	85
Operation	85
XGLC Link Aggregation.....	86
Link Aggregation Control.....	86
Redundancy Options.....	87
Distribution Options.....	87
Horizontal Link Aggregation with Two Ethernet Switches.....	88
Link Aggregation Status	88
Configuring Management Settings	89
ORBEM and the Web Element Manager.....	90
Configuring ORBEM Client and Port Parameters.....	91
Configuring Internet Inter-ORB Protocol (IOP) Transport Parameters.....	92
Verifying ORBEM Parameters	93

SNMP Support.....	94
Configuring SNMP and Alarm Server Parameters.....	94
Verifying SNMP Parameters	95
Controlling SNMP Trap Generation.....	97
Verifying and Saving Your Configuration	99
Verifying the Configuration	100
Feature Configuration.....	100
Service Configuration.....	101
Context Configuration.....	101
System Configuration	101
Finding Configuration Errors	102
Saving the Configuration on the ASR 5000 Chassis.....	103
System Interface and Port Configuration Procedures.....	105
Creating Contexts.....	106
Viewing and Verifying Contexts	106
Creating and Configuring Ethernet Interfaces and Ports	107
Creating an Interface.....	107
Configuring a Port and Binding It to an Interface	108
Configuring a Static Route for an Interface.....	108
Viewing and Verifying Port Configuration	109
Creating and Configuring ATM Interfaces and Ports	111
Enabling the OLC (ATM) Line Card	111
Creating an IP Interface for Use with an ATM Port.....	112
Configuring an ATM Port to Use an IP Interface.....	112
Configuring an ATM Port for an SS7 Link.....	113
Binding an SS7 Link to an ATM Port.....	113
Verifying Port and Interface Configuration	113
Creating and Configuring Frame Relay Interfaces and Ports	115
Setting the Characteristics of the Channelized Line Card	115
Configuring the Channel Characteristics.....	116
Binding a DLCI	116
Verifying the Frame Relay Interface Configuration and Status.....	117
Display Port and DLCI Configuration Details.....	117
Display Port and DLCI Configuration and Status	118
Software Management Operations.....	119
Understanding the Local File System.....	120
File Types Used by the Local File System	120
Understanding the boot.sys File	121
Maintaining the Local File System.....	122
File System Management Commands.....	122
Synchronizing the File System	122
Creating Directories	123
Renaming Files and Directories.....	123
Copying Files on the ASR 5000 Chassis.....	123
Deleting Files	125
Deleting Directories.....	125
Formatting Local Devices	126
Applying Pre-existing CLI Configuration Files.....	126
Viewing Files on the Local File System.....	127
Viewing the Contents of a Local Device	127
Viewing CLI Configuration and boot.sys Files	128
Validating an Operating System File	128

Configuring the Boot Stack	129
System Boot Methods	129
Viewing the Current Boot Stack	129
Adding a New Boot Stack Entry	130
Deleting a Boot Stack Entry	132
Network Booting Configuration Requirements	132
Configuring the Boot Interface	132
Configuring the Boot Network	133
Configuring Boot Network Delay Time	134
Configuring a Boot Nameserver	135
Upgrading the Operating System Software	136
Identifying OS Release Version and Build Number	136
Saving a Copy of the Current Configuration File	136
Preparing for a Software Downgrade	136
Software Upgrade Methods	137
On-Line Software Upgrade	137
System Requirements to Support the On-line Software Upgrade Method	138
CLI Verification and System Preparation	138
Stage 1 - Soft Busy-out	138
Stage 2 - Stand-alone Operation	139
Stage 3 - Management Card Upgrade	139
Stage 4 - Reboot All Packet Processing Cards	140
Stage 5 - Return System to Normal Operation	140
Performing an On-line Software Upgrade	140
Aborting an On-line Software Upgrade	144
Restoring the Previous (Pre-online Upgrade) Software Image	144
Off-line Software Upgrade	145
Restoring the Previous Software Image	148
Managing License Keys	149
New System License Keys	149
Session Use and Feature Use Licenses	149
Installing New License Keys	150
Cutting and Pasting the Key	150
Adding License Keys to Configuration Files	151
License Expiration Behavior	151
Requesting License Keys	153
Viewing License Information	154
Deleting a License Key	155
Management Card Replacement and License Keys	156
Managing Local-User Administrative Accounts	157
Configuring Local-User Password Properties	157
Configuring Local-User Account Management Properties	157
Local-User Account Lockouts	157
Local-User Account Suspensions	158
Changing Local-User Passwords	158
Monitoring the System	159
SNMP Notifications	160
Monitoring System Status and Performance	160
Clearing Statistics and Counters	162
Monitoring ASR 5000 Hardware Status	163
Configuring and Maintaining Bulk Statistics	165
Configuring Communication With the Collection Server	166
Configuring Standard Settings	166

Configuring Optional Settings	166
Configuring Bulk Statistic Schemas	167
Verifying Your Configuration	167
Saving Your Configuration	169
Viewing Collected Bulk Statistics Data	170
Manually Gathering and Transferring Bulk Statistics	172
Clearing Bulk Statistics Counters and Information	173
Bulk Statistics Event Log Messages	174
Configuring and Viewing System Logs	175
System Log Types	176
Configuring Event Logging Parameters	177
Configuring Event Log Filters	177
Configuring syslog Servers	177
Configuring Trace Logging	179
Configuring Active Logs	180
Configuring Monitor Logs	181
Enabling Monitor Logs	181
Disabling Monitor Logs	181
Viewing Logging Configuration and Statistics	182
Viewing Event Logs Using the CLI	183
Configuring and Viewing Software Crash Logging Parameters	184
Configuring Software Crash Log Destinations	184
Viewing Abridged Crash Logs Using the CLI	185
Saving Log Files	188
Event ID Overview	189
Event Severities	193
Understanding Event ID Information in Logged Output	193
Troubleshooting the System	195
Detecting Faulty Hardware	196
Using the CLI to View Status LEDs	196
Checking the LED on the PFU	197
Checking the LEDs on the SMC	198
SMC Run/Fail LED States	199
SMC Active LED States	201
SMC Standby LED States	202
SMC Status LED States	202
SMC Service LED States	203
SMC Busy LED States	204
Checking the LEDs on the Packet Processing Cards	204
Packet Processing Card Run/Fail LED States	205
Packet Processing Card Active LED States	207
Packet Processing Card Standby LED States	208
Checking the LEDs on the SPIO	209
SPIO Run/Fail LED States	211
SPIO Active LED States	211
SPIO Standby LED States	212
SPIO Interface Link LED States	213
SPIO Interface – Activity LED States	213
Checking the LEDs on Ethernet Line Cards	214
Ethernet Line Card Run/Fail LED States	214
Ethernet Line Card Active LED States	215
Ethernet Line Card Standby LED States	216
Ethernet Line Card Interface – Link LED States	216

Ethernet Line Card Interface Activity LED States	217
Checking the LEDs on the RCC	218
RCC Run/Fail LED States	218
RCC Active LED States	219
RCC Standby LED States	220
Testing System Alarm Outputs	221
Taking Corrective Action	222
Manually Initiating a Management Card Switchover	222
Manually Initiating a Packet Processing Card Migration	222
Manually Initiating a Line Card or SPIO Switchover	223
Halting Cards	224
Initiate a Card Halt	224
Restoring a Previously Halted Card	225
Verifying Network Connectivity	226
Using the ping Command	226
Using the traceroute Command	228
Viewing IP Routes	229
Viewing the Address Resolution Protocol Table	229
Using the System Diagnostic Utilities	231
Using the Monitor Utility	231
Using the Protocol Monitor	231
Using the Protocol Monitor for a Specific Subscriber	232
Using the DHCP Testing Tool	235
System Recovery	237
Prerequisites	238
Console Access	238
Boot Image	238
Accessing the boot CLI	239
Initiate a Reboot	239
Interrupt the Boot Sequence	239
Enter CLI Mode	240
boot Command Syntax	240
Bootting from a Selected Image	241
Boot Using No Configuration File	241
Boot Using A Specified Configuration File	241
Engineering Rules	243
CLI Session Rules	244
ASR 5000 Interface and Port Rules	244
Line Card Rules	244
Packet Data Network (PDN) Interface Rules	245
ASR 5000 Packet Processing Card Rules	246
Context Rules	247
Subscriber Rules	248
Service Rules	248
Access Control List (ACL) Engineering Rules	249
System Software Task and Subsystem Descriptions	251
Overview	252
Primary Task Subsystems	253
Primary Subsystem Controllers and Managers	255
ASR 5x00 Subsystems	256
Access Control Lists	283
Overview	284

Understanding ACLs	285
Rule(s)	285
Actions	285
Criteria	285
Rule Order	286
Configuring ACLs on the System	287
Creating ACLs	287
Configuring Action and Criteria for Subscriber Traffic	288
Configuring an	288
Verifying the ACL Configuration	289
Applying IP ACLs	290
Applying an ACL to an Individual Interface	291
Applying ACL to Interface	292
Verifying the ACL Configuration on Interface	292
Applying an ACL to All Traffic Within a Context	293
Applying ACL to Context	293
Verifying the ACL Configuration in a Context	294
Applying an ACL to a RADIUS-based Subscriber	295
Applying an ACL to an Individual Subscriber	295
Verifying the ACL Configuration to an Individual Subscriber	296
Applying a Single ACL to Multiple Subscribers	297
Applying an ACL to the Subscriber Named default	298
Applying an ACL to Service-specified Default Subscribers	300
Congestion Control	305
Overview	306
Configuring Congestion Control	307
Configuring the Congestion Control Threshold	307
Configuring Service Congestion Policies	307
Configuring Overload Reporting on the MME	308
Enabling Congestion Control Redirect Overload Policy	308
Verify the Service Overload Policies	309
Verify the Congestion Control Configuration	309
Disconnecting Subscribers Based on Call or Inactivity Time	311
Content Service Steering	313
Overview	314
Configuring Internal Content Service Steering	315
Defining IP Access Lists for Internal CSS	315
Applying an ACL to an Individual Subscriber (Optional)	316
Applying an ACL to Multiple Subscribers (Optional)	316
Applying an ACL to the Subscriber Named default (Optional)	316
Applying an ACL to Service-specified Default Subscribers (Optional)	316
Applying an ACL to Multiple Subscribers via APNs (Optional)	316
Interchassis Session Recovery	317
Overview	318
Interchassis Communication	318
Checkpoint Messages	318
AAA Monitor	318
BGP Interaction	319
Requirements	319
ICSR Operation	321
Chassis Initialization	324
Chassis Operation	324

Chassis Communication	324
Chassis Switchover	324
Configuring Interchassis Session Recovery (ICSR)	325
Configuring the Service Redundancy Protocol (SRP) Context	326
Creating and Binding the SRP Context	326
Configuring the SRP Context Parameters	326
Configuring the SRP Context Interface Parameters	327
Verifying SRP Configuration	328
Modifying the Source Context for ICSR	329
Configuring BGP Router and Gateway Address	329
Configuring SRP Context for BGP	329
Verifying BGP Configuration	330
Modifying the Destination Context for ICSR	330
Configuring BGP Router and Gateway Address in Destination Context	330
Configuring SRP Context for BGP for Destination Context	330
Setting Subscriber to Default Mode	331
Verifying BGP Configuration in Destination Context	331
Disabling Bulk Statistics Collection on a Standby System	331
Verifying the Primary and Backup Chassis Configuration	332
QoS Management	333
Introduction	334
Dynamic QoS Renegotiation	334
How Dynamic QoS Renegotiation Works	334
Initial QoS	334
Service Detection	335
Classification of Application Traffic	335
QoS Renegotiation for a Subscriber QoS Profile	335
Network Controlled QoS (NCQoS)	337
How Network Controlled QoS (NCQoS) Works	337
Configuring Dynamic QoS Renegotiation	338
Configuring ACL for Dynamic QoS Renegotiation	338
Configuring Charging Action for Dynamic QoS Renegotiation	339
Configuring Rulebase for Dynamic QoS Renegotiation	339
Configuring APNs for Dynamic QoS Renegotiation	339
Configuring Network Controlled QoS (NCQoS)	341
Configuring Packet Filter for NCQoS	341
Configuring Charging Action for NCQoS	342
Configuring APN for NCQoS	342
Monitoring Dynamic QoS Renegotiation Operation	343
Event IDs Pertaining to Dynamic QoS Renegotiation	344
RADIUS Attributes	344
Routing	345
Routing Policies	346
Creating IP Prefix Lists	346
Creating Route Access Lists	346
Creating AS Path Access Lists	347
Creating Route Maps	347
Sample Configuration	347
Static Routing	349
Adding Static Routes to a Context	349
Deleting Static Routes From a Context	350
OSPF Routing	351
OSPF Version 2 Overview	351

Basic OSPFv2 Configuration	352
Enabling OSPF Routing For a Specific Context	352
Enabling OSPF Over a Specific Interface	352
Redistributing Routes Into OSPF (Optional)	352
Confirming OSPF Configuration Parameters	353
OSPFv3 Routing	354
OSPFv3 Overview	354
Basic OSPFv3 Configuration	354
Enabling OSPFv3 Routing For a Specific Context	354
Enabling OSPFv6 Over a Specific Interface	354
Redistributing Routes Into OSPFv3 (Optional)	355
Confirming OSPFv3 Configuration Parameters	355
Viewing Routing Information	356
Equal Cost Multiple Path (ECMP)	357
BGP-4 Routing	358
Overview of BGP Support	358
Configuring BGP	359
Redistributing Routes Into BGP (Optional)	359
Bidirectional Forwarding Detection	360
Overview of BFD Support	360
Configuring BFD	360
Configuring a BFD Context	360
Configuring IPv4 BFD for Static Routes	361
Configuring BFD for BGP Single Hop	361
Configuring Multihop BFD for BGP	362
Scaling of BFD	362
Associate BGP Neighbors with the Context	363
Associate BFD Neighbor Groups with the BFD Protocol	363
Session Recovery	365
How Session Recovery Works	366
Additional Hardware Requirements	368
Configuring the System to Support Session Recovery	369
Enabling Session Recovery	369
Enabling Session Recovery on an Out-of-Service System	369
Enabling Session Recovery on an In-Service System	371
Disabling the Session Recovery Feature	373
Viewing Session Recovery Status	373
Viewing Recovered Session Information	374
VLANs	379
Overview	380
Creating VLAN Tags	380
Verify the Port Configuration – ASR 5000	381
Configuring Subscriber VLAN Associations	383
RADIUS Attributes Used	383
Configuring Local Subscriber Profiles	383
Verify the Subscriber Profile Configuration	383

About this Guide

This preface describes the *System Administration Guide*, how it is organized and its document conventions.

The *System Administration Guide* describes how to generally configure and maintain StarOS running on an ASR 5000 platform. It also includes information on monitoring system performance and troubleshooting.

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)	Warns you to take proper grounding precautions before handling ESD sensitive components or devices.

Typeface Conventions	Description
Text represented as a <i>screen display</i>	This typeface represents displays that appear on your terminal screen, for example: <i>Login:</i>
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 <i>variable</i> }	Required keywords and variables are surrounded by grouped braces. Required keywords and variables are those components that are required to be entered as part of the command syntax.
[keyword or <i>variable</i>]	Optional keywords or variables, or those that a user may or may not choose to use, are surrounded by brackets.

Command Syntax Conventions	Description
	Some commands support alternative variables. These options are documented within braces or brackets by separating each variable with a vertical bar. These variables can be used in conjunction with required or optional keywords or variables. For example: <pre>{ nonce timestamp }</pre> OR <pre>[count <i>number_of_packets</i> size <i>number_of_bytes</i>]</pre>

Supported Documents and Resources

Related Documentation

The most up-to-date information for this product is available in the product *Release Notes* provided with each software release.

The following user documents are available on www.cisco.com:

- *ASR 5000 Installation Guide*
- *Command Line Interface Reference*
- *AAA and GTPP Interface Administration Reference*
- *Product Overview*
- *Release Change Reference*
- *Statistics and Counters Reference*
- *SNMP MIB Reference*
- *Thresholding Configuration Guide*
- *Cisco Web Element Manager Installation and Administration Guide*
- Product-specific and feature-specific Administration guides

Obtaining Documentation

The most current Cisco documentation is available on the following website:

<http://www.cisco.com/cisco/web/psa/default.html>

Use the following path selections to access user documentation:

Products > Wireless > Mobile Internet> Network Functions

Contacting Customer Support

Use the information in this section to contact customer support.

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 are required to access this site. Please contact your Cisco sales or service representative for additional information.

Chapter 1

Understanding System Operation and Configuration

The ASR 5000 system provides wireless carriers with a flexible solution that can support a wide variety of services. These services are described in detail in the *ASR 5000 Product Overview Guide*.

Before you connect to the command line interface (CLI) and begin system configuration, you must understand how the system supports these services. This chapter provides terminology and background information to consider before you configure the system. The following sections are included:

- [Terminology](#)
- [How the System Selects Contexts](#)
- [Understanding the ASR 5000 Boot Process](#)
- [IP Address Notation](#)
- [Alphanumeric Strings](#)

Terminology

This section defines important terms used in the remaining chapters of this guide.

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 context is configured and operates independently of the others. Once a context has been created, administrative users can configure services, logical IP interfaces, and subscribers for that context and then bind the logical interfaces to physical ports.

You can also assign a domain alias to a context; if a subscriber's domain name matches one of the configured alias names for a context, that context is used.

Ports

Ports are the physical connectors on line cards that support remote access and subscriber traffic. Port configuration includes 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 SPIO 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).

For complete information on line cards and port assignments, refer to the *ASR 5000 Installation and Administration Guide*.

Logical Interface

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 the 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 command line interface (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:

- Gateway GPRS Support Node (GGSN) services
- Serving GPRS Support Node (SGSN) Services
- Packet Data Serving Node (PDSN) services
- Home Agent (HA) services
- Home Agent (HA) services
- Dynamic Host Control Protocol (DHCP) services
- Access Service Network Gateway (ASN-GW) services
- ASN Paging Controller and Location Registry services
- Packet Data Interworking Function (PDIF) services
- Session Control Manager (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
- Intelligent Policy Control Function (IPCF) Services (PCC-Service, PCC-Policy, PCC-AF)

AAA Servers

Authentication, Authorization and Accounting (AAA) servers store profiles, perform authentication, and maintain accounting records for each mobile data subscriber. The AAA servers communicate with the system over an AAA interface. The system supports the configuration of up to 128 interfaces to AAA servers.

It is important to note that for Mobile IP, there can be Foreign AAA (FAAA) and Home AAA (HAAA) servers. FAAA servers typically reside in the carrier's network. HAAA servers 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 an AAA proxy server.



Important: Mobile IP support depends on the availability and purchase of a standalone license or a license bundle that includes Home Agent (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 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, or TACACS+.

How the System Selects Contexts

This section describes the process that determines which context to use for context-level administrative users or subscriber sessions. Understanding this process allows you to better plan your configuration in terms of how many contexts and interfaces you need to 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.

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 an 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.

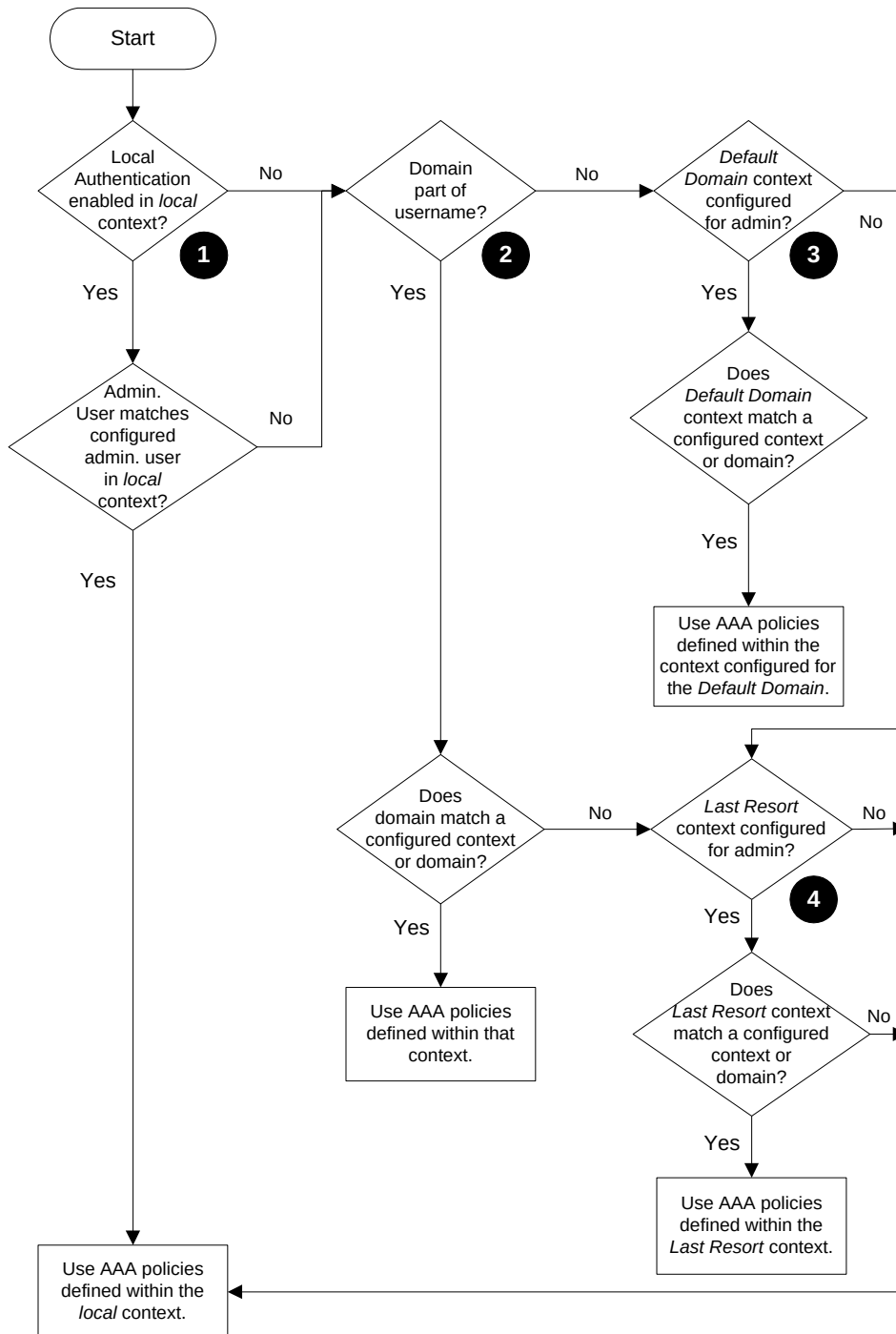
The following table and flowchart describe the process that the system uses to select an AAA context for a context-level administrative user. Items in the table correspond to the circled numbers in the flowchart.

Table 1. Context-level Administrative User AAA Context Selection

Item	Description
1	During authentication, the system determines whether local authentication is enabled in the <i>local</i> context. If it is, the system attempts to authenticate the administrative user in the <i>local</i> 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 <i>local</i> context. If not, proceed to item 2 in this table.
2	If local authentication is disabled on the system or if the administrative user's username is not configured in the <i>local</i> 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.

Item	Description
3	If there was no domain specified in the username or the domain is not recognized, the system determines whether an <i>AAA Administrator Default Domain</i> is configured. If the default domain is configured and it matches a configured context, the AAA configuration within the <i>AAA Administrator Default Domain</i> context is used. If the default domain is not configured or does not match a configured context or domain, go to item 4 item below.
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 <i>AAA Administrator Last Resort context parameter</i> 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 <i>local</i> context is used.

Figure 1. Context-level Administrative User AAA Context



Context Selection for Subscriber Sessions

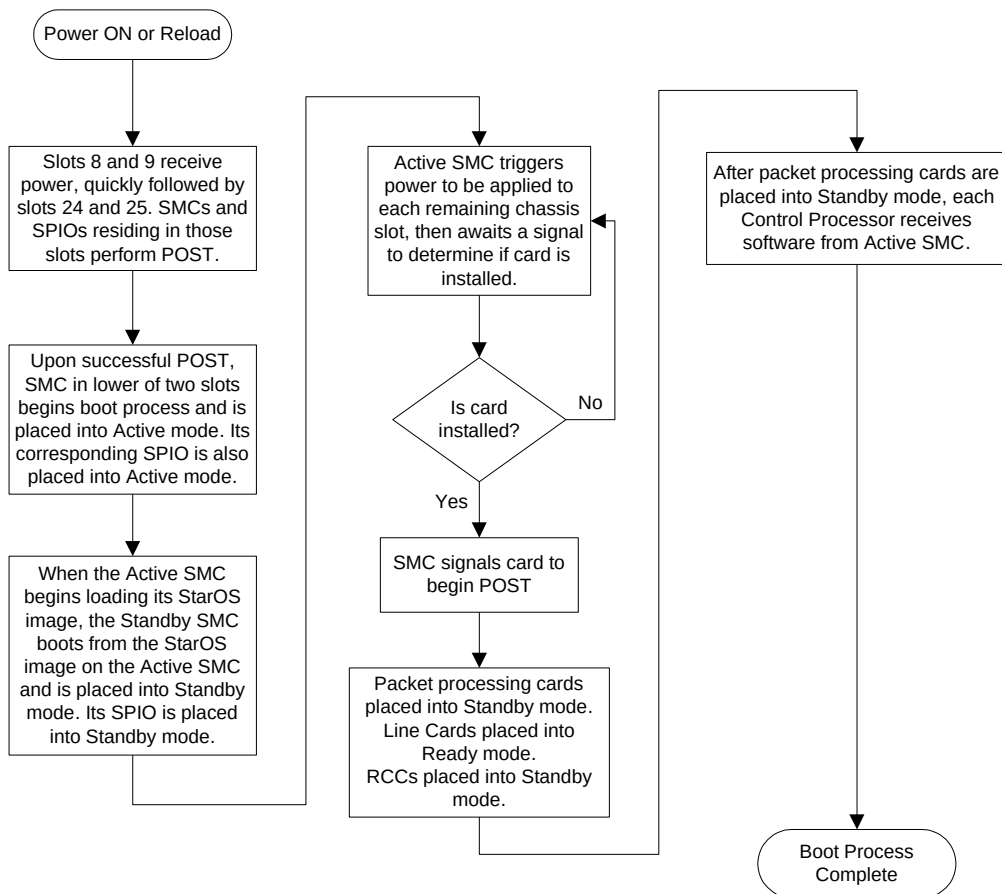
The context selection process for a subscriber session is more involved than that for the administrative users. Subscriber session context selection information for specific products is located in the *Administration Guide* for the individual product.

Understanding the ASR 5000 Boot Process

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 2. ASR 5000 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 SMC slots (slots 8 and 9) receive power. Therefore, the 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 SMC in slot 8 successfully executes all POSTs, the card in slot 8 becomes the active SMC. The SMC in slot 9 becomes the standby card. If there is a problem with the SMC in slot 8, the card in slot 9 becomes the active 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 SMCs.
- Step 4** The active 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 SMC CompactFlash. The standby SMC observes the active card startup. If the file on the active card is loaded normally, the standby SMC boots from the active card image. If the active 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 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 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 PSCs and line cards installed in the system, they each perform their own series of POSTs.
- Step 7** After successful POST, each of the PSCs enter standby mode.
- Step 8** Installed line cards remain in steady mode until their corresponding PSC is made active via configuration. After the 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 standby mode.
- Step 9** After entering the standby mode, each of the PSC control processors (CPs) communicate with the SMC to receive the appropriate code.
- Step 10** 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 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. Using a file for offline system configuration 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.

 **Important:** Always save configuration files in UNIX format. Failure to do so can result in errors that prevent configuration file 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 the *Software Management Operations* chapter.
- 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 the *Software Management Operations* chapter.

 **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.
 - **/flash**: a solid-state device with limited storage.
- 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.

IP Address Notation

When configuring a port interface via the CLI you must enter an IP address. The CLI always accepts an IPv4 address, and in some cases accepts an IPv6 address as an alternative.

For some configuration commands, the CLI also accepts CIDR notation. Always view the online Help for the CLI command to verify acceptable forms of IP address notation.

IPv4 Dotted-Decimal Notation

An Internet Protocol Version 4 (IPv4) address consists of 32 bits divided into four octets. These four octets are written in decimal numbers, ranging from 0 to 255, and are concatenated as a character string with full stop delimiters (dots) between each number.

For example, the address of the loopback interface, usually assigned the host name localhost, is 127.0.0.1. It consists of the four binary octets 01111111, 00000000, 00000000, and 00000001, forming the full 32-bit address.

IPv4 allows 32 bits for an Internet Protocol address and can, therefore, support 4,294,967,296 addresses

IPv6 Colon-Separated Notation

An Internet Protocol Version 6 (IPv6) address has two logical parts: a 64-bit network prefix, and a 64-bit host address part. An IPv6 address is represented by eight groups of 16-bit hexadecimal values separated by colons (:).

A typical example of a full IPv6 address is 2001:0db8:85a3:0000:0000:8a2e:0370:7334

The hexadecimal digits are case-insensitive.

The 128-bit IPv6 address can be abbreviated with the following rules:

- Leading zeroes within a 16-bit value may be omitted. For example, the address fe80:0000:0000:0202:b3ff:fe1e:8329 may be written as fe80:0:0:0:202:b3ff:fe1e:8329
- One group of consecutive zeroes within an address may be replaced by a double colon. For example, fe80:0:0:0:202:b3ff:fe1e:8329 becomes fe80::202:b3ff:fe1e:8329

IPv6 allows 128 bits for an Internet Protocol address and can support 340,282,366,920,938,000,000,000,000,000,000,000,000 internet addresses.

CIDR Notation

Classless Inter-Domain Routing (CIDR) notation is a compact specification of an Internet Protocol address and its associated routing prefix. It is used for both IPv4 and IPv6 addressing in networking architectures.

CIDR is a bitwise, prefix-based standard for the interpretation of IP addresses. It facilitates routing by allowing blocks of addresses to be grouped into single routing table entries. These groups (CIDR blocks) share an initial sequence of bits in the binary representation of their IP addresses.

CIDR notation is constructed from the IP address and the prefix size, the latter being the number of leading 1 bits of the routing prefix. The IP address is expressed according to the standards of IPv4 or IPv6. It is followed by a separator character, the slash (/) character, and the prefix size expressed as a decimal number.

The address may denote a single, distinct, interface address or the beginning address of an entire network. In the latter case the CIDR notation specifies the address block allocation of the network. The maximum size of the network is given by the number of addresses that are possible with the remaining, least-significant bits below the prefix. This is often called the host identifier.

For example:

- the address specification 192.168.100.1/24 represents the given IPv4 address and its associated routing prefix 192.168.100.0, or equivalently, its subnet mask 255.255.255.0.
- the IPv4 block 192.168.0.0/22 represents the 1024 IPv4 addresses from 192.168.0.0 to 192.168.3.255.
- the IPv6 block 2001:DB8::/48 represents the IPv6 addresses from 2001:DB8:0:0:0:0:0:0 to 2001:DB8:0:FFFF:FFFF:FFFF:FFFF:FFFF.
- ::1/128 represents the IPv6 loopback address. Its prefix size is 128, the size of the address itself, indicating that this facility consists of only this one address.

The number of addresses of a subnet defined by the mask or prefix can be calculated as 2, in which the address size for IPv4 is 32 and for IPv6 is 128. For example, in IPv4, a mask of /29 gives 8 addresses.

Alphanumeric Strings

Some CLI commands require the entry of an alphanumeric string to define a value. The string is a contiguous collection of alphanumeric characters with a defined minimum and maximum length (number of characters).

Character Set

The alphanumeric character set is a combination of alphabetic (Latin letters) and/or numeric (Arabic digits) characters. The set consists of the numbers 0 to 9, letters A to Z (uppercase) and a to z (lowercase). The underscore character (`_`) and dash/hyphen (`-`) are also considered to be members of the alphanumeric set of characters.

Blank spaces (whitespaces or SPACE characters) should mostly be avoided in alphanumeric strings, except in certain ruledef formats, such as time/date stamps.

Do not use any of the following “special” characters in an alphanumeric string except as noted below:

- `&` (ampersand)
- `'` (apostrophe)
- `<>` (arrow brackets) [see exception below]
- `*` (asterisk) [see wildcard exception below]
- `{ }` (braces)
- `[ ]` (brackets)
- `$` (dollar sign) [see wildcard exception below]
- `!` (exclamation point) [see exception below]
- `( )` [parentheses]
- `%` (percent) [see exception below]
- `#` (pound sign) [see exception below]
- `?` (question mark)
- `'` (quotation mark – single)
- `"` (quotation mark – double)
- `;` (semicolon)
- `\` (slash – backward) [see exception below]
- `/` (slash – forward) [see exception below]
- `~` (tilde)
- `|` (vertical bar) [see exception below]

The following characters may appear in strings entered in ruledefs, APNs, license keys and other configuration/display parameters:

- `<>` (arrow brackets) [less than or greater than]
- `*` (asterisk) [wildcard]
- `:` (colon)
- `$` (dollar sign) [wildcard]

- . (dot)
- = (equals sign)
- ! (exclamation point)
- % (percent)
- / (slash – forward)
- | (vertical bar)

The following characters may be used to delimit the domain from the user name for global AAA functions:

- @ (at sign)
- - (dash or hyphen)
- # (hash or pound sign)
- % [percent]
- \ (slash – backward) [must be entered as double slash “\\”]
- / (slash – forward)

Quoted Strings

If descriptive text requires the use of spaces between words, the string must be entered within double quotation marks (“”). For example:

```
interface "Rack 3 Chassis 1 port 5/2"
```


Chapter 2

Getting Started

Following successful installation of the system hardware, you must configure a set of software parameters. You then save these settings in a system configuration file that is launched whenever the system is reloaded.

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

- [ASR 5000 Configuration](#)
- [Using the ASR 5000 Quick Setup Wizard](#)
- [Using the CLI for Initial Configuration](#)
- [Configuring the System for Remote Access](#)
- [Configuring the Management Interface with a Second IP Address](#)

ASR 5000 Configuration

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

The serial console port is located on the SPIO card installed in slot 24. This wizard is guides you through the initial configuration of the system.

You can choose not to use the wizard and perform the initial configuration by issuing commands via the command line interface (CLI). You can manually launch the wizard by running the **setup** command in the Exec mode. Refer to the *Command Line Interface Reference* for details.

The following sections describe how to configure the system.

Using the ASR 5000 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
- 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 run logic of the wizard along with additional information and notes.

Figure 3. ASR 5000 Quick Setup Wizard Logic Diagram

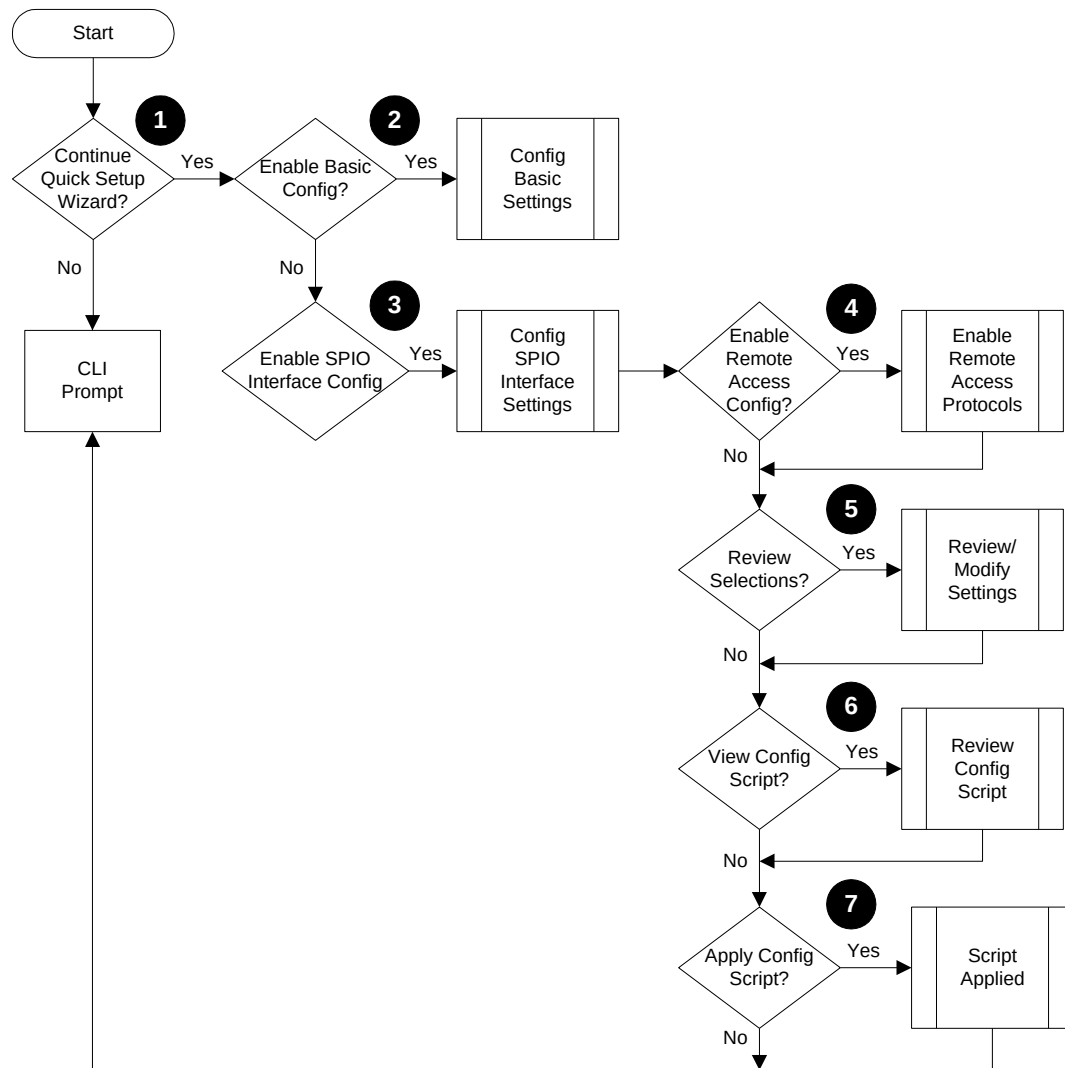
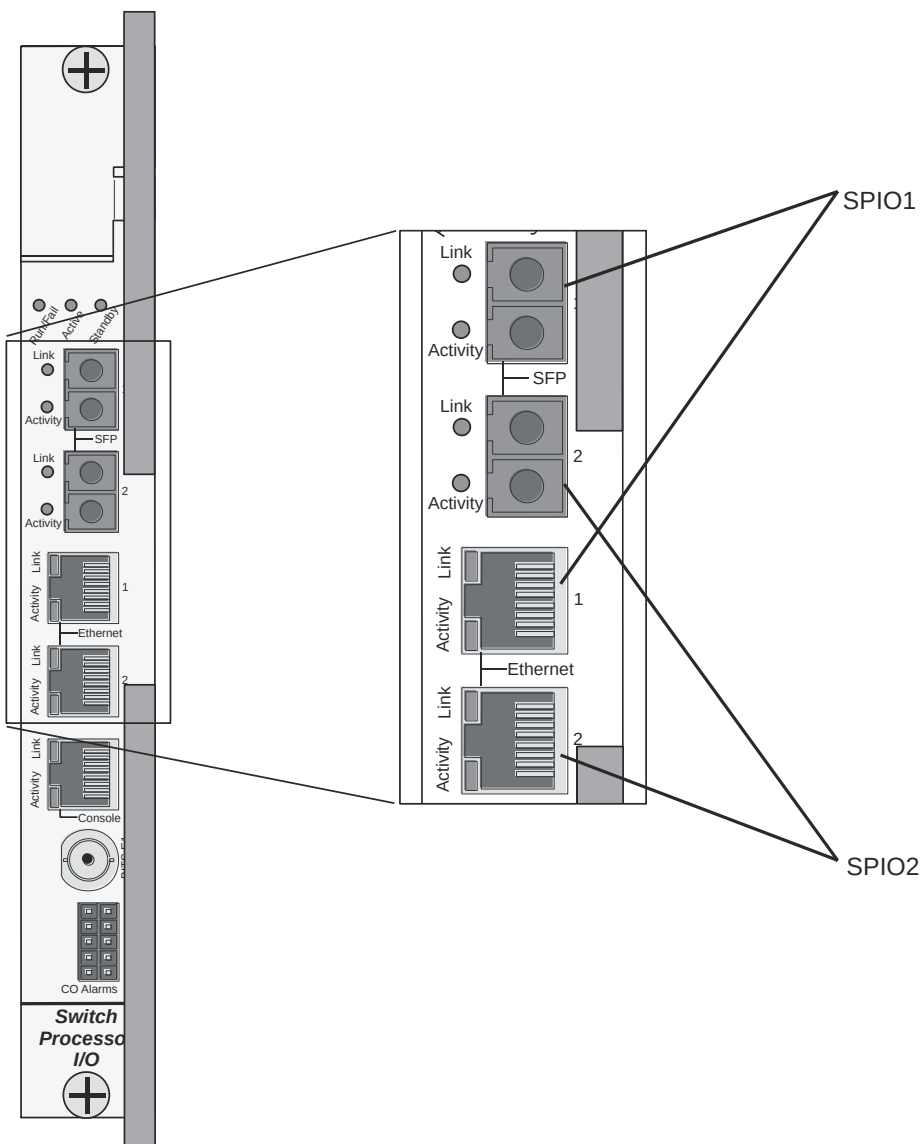


Table 2. Quick Setup Wizard Logic Diagram Callout Descriptions

Item	Description/Notes
1	Enter or exit the wizard. - Enter no at the prompt to automatically be directed to the command line interface (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.
2	Configure an administrative username/password and a hostname for the system. - The name of the default administrative user configured through the wizard is <i>admin</i>. - Administrative user name is an alphanumeric string of 1 through 32 characters that is case sensitive. - Administrative user password is an alphanumeric string of 1 through 63 characters that is case sensitive. - Configure a valid, non-null hostname. The hostname is an alphanumeric string of 1 through 63 characters that is case sensitive.
3	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. <i>SPIO1</i> 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 <i>rj-45</i>. - 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 <i>Configuring System Settings</i>.
4	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. For maximum security, it is recommended that you use only SSH v2.

Item	Description/Notes
5	Review and/or modify the configuration of previous prompts. 1. Enter the number of the prompt to be modified. 2. Configure the parameter. 3. <i>Optional.</i> Repeat <i>step 1</i> and <i>step 2</i> to modify additional settings. 4. Enter “done” when you have completed all changes.
6	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 (<i>variable</i>).
7	Apply the configuration file to the system. Once applied, the parameter configuration is automatically saved to the system.cfg file stored on the primary SMC compact flash card.

Figure 4. 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 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 CLI prompt, enter:

```
[local]host_name# configure

[local]host_name(config)#
```

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

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

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

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.

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
<i>name</i>	Specifies the security administrator's name as an alphanumeric string of 1 through 32 characters that is case sensitive.
password <i>password</i>	Specifies the password for the security administrator as an alphanumeric string of 1 through 63 characters that is case sensitive.
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 flash memory.
no-cli	Specifies that the security administrator cannot access the system command line interface (CLI). 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 (in seconds) that the operator can maintain a session with the system. as an integer from 0 through 300000000. Default: 0 If the absolute timeout value is exceeded, the operator session is automatically terminated.
timeout-idle	Specifies the maximum amount of time (in seconds) that an operator session can remain idle before being automatically terminated. This is an integer from 0 through 300000000. Default: 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:

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

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

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

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

host_name is the name by which the system will be recognized on the network. The hostname is an alphanumeric string of 1 through 63 characters that 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:

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

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

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

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

interface_name is the name of the interface expressed as an alphanumeric string of 1 through 79 characters that 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 | ipv6 address } ipaddress subnetmask
```

Variable	Description
IPv4 – ip address	
<i>ipaddress</i>	Specifies the IP address for the interface in IPv4 dotted-decimal format.
<i>subnetmask</i>	Specifies the subnet mask for the interface in IPv4 dotted-decimal or CIDR notation.
IPv6 – ipv6 address	
<i>ipaddress</i>	Specifies the IP address for the interface in IPv6 colon-separated format.
<i>subnetmask</i>	Specifies the subnet mask for the interface in IPv6 colon-separated or CIDR notation.



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:

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

```
[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 | ipv6 } route gw_address interface_name
```

Variable	Description
<i>gw_address</i>	Specifies the IP address of the default gateway in IPv4 dotted-decimal format or IPv6 colon-separated format.
<i>interface_name</i>	Specifies the name of the interface that was configured in step 7b.

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

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

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

Step g.....Enter the Ethernet Port Configuration mode:

```
port ethernet slot#/port#
```

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

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:

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

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

```
[local]host_name(config-port-slot#/port#)# 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 | 10000 } 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 = 10 Mbps • 100 = 100 Mbps • 1000 = 1000 Mbps Use 1000 Mbps only for the SFP ports on the GELC/GLC2, QGLC or SPIO Line Cards. 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. 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:

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

```
[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:** If there are two simultaneous telnet sessions, and one administrator deletes the context into which the other administrator is logged, the administrator in the deleted context will not be automatically kicked into the *local* context. Although the deleted context will still appear in the CLI prompt, context specific commands will generate errors.

 **Important:** For maximum security, use SSH v2.

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

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

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

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

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

```
[local]host_name(config-ctx)# ssh generate key [ type { v1-rsa | v2-rsa | v2-dsa } ]
```

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

```
[local]host_name(config-ctx)# server sshd
[local]host_name(config-sshd)# subsystem sftp
[local]host_name(config-sshd)# exit
```

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

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

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

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



```
[local]host_name#
```

Step 6 Verify the configuration by entering the following command:

```
[local]host_name# 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

  subsystem sftp

  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 7 Verify the configuration of the IP routes by entering the following command:

```
[local]host_name# 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 spiol
*<network> 0.0.0.0 connected 0 0 spiol
```

Step 8 Verify the interface binding by entering the following command:

```
[local]host_name# 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: spiolIntf 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 9 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring the 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:

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

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

```
[local]host_name(config)# context local  
[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  
[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 | ipv } address ipaddress subnet_mask  
secondary
```

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

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

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

```
[local]host_name# 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 3

Configuring System Settings

This chapter provides instructions for configuring the following system options:

- [Configuring a Second Management Interface](#)
- [Configuring System Timing](#)
- [Configuring Transmit Timing Source](#)
- [Enabling CLI Timestamping](#)
- [Configuring System Administrative Users](#)
- [Configuring TACACS+ for System Administrative Users](#)
- [Configuring a Chassis Key](#)
- [Configuring Virtual MAC Addresses](#)
- [Configuring Packet Processing and Line Card Availability](#)
- [Configuring Line Card and SPIO Port Redundancy](#)
- [Configuring ASR 5000 Link Aggregation](#)

It is assumed that the procedures to initially configure the system as described in the *Getting Started* chapter have been completed.



Important: The commands used in the configuration examples in this section are the 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 Switch Processor Input/Output (SPIO) card. This section provides described how to configure 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 (slot number 24 or 25).
- Enter IP addresses using IPv4 dotted-decimal or IPv6 colon-separated notation.
- For **port ethernet port#**, use the physical port on the SPIO card that will be used. 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 (copper) and dual SFP (optical fiber) 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.
- In the { **ip** | **ipv6** } **route** command, other keyword options, instead of the gateway IP address, are available and include: **next-hop** IP address, **point-to-point**, and **tunnel**.

Verifying and Saving Your Interface and Port Configuration

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

```
show ip interface
```

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

```
Intf Name: mgmt2

Intf Type:          Broadcast

Description:        management2

VRF:                None

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: 60 secs

L3 monitor LC-port switchover: Disabled

Number of Secondary Addresses: 0
```

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; it 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
```

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

Configuring System Timing

The system is equipped with a clock that supplies the timestamp for statistical 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) 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:

- Enter the date and time in the format YYYY:MM:DD:HH:mm or YYYY:MM:DD:HH:mm:ss.
- Refer to the online Help for the **clock timezone** command 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 that 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.

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.
- Enter the IP address of NTP servers using IPv4 dotted-decimal or IPv6 colon-separated notation.

 **Important:** Configure the system with at least two NTP servers. It is recommend that four servers are configured.

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

Verifying the NTP Configuration

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 parameters output by the **show ntp associations** command.

Column Title	Description
remote	List of 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	Last reported NTP reference to which the server is synchronizing.
st	NTP server stratum level.
t	Communication type: broadcast, multicast, etc.
when	Number of seconds since the last contact.
poll	Polling interval between the system and the NTP server.
reach	Octal value of the reachability shift register indicating which responses were received for the previous eight polls to this NTP server.

Column Title	Description
delay	Round-trip delay (in milliseconds) for messages exchanged between the system and the NTP server.
offset	Number of milliseconds by which the system clock must be adjusted to synchronize it with the NTP server.
jitter	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 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) and line-timing.

BITS-timing uses Stratum 3-compliant BITS modules resident on the SPIOs.

Line-timing recovers the receive timing from an external clock through a specified port on an Optical or Channelized line card (OLC/OLC2 or CLC/CLC2).

The timing is then distributed via the SPIO to all line cards in the chassis.

 **Important:** To use BITS-timing, the SPIO card must include the optional BITS BNC or 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 assures redundancy. When enabled BITS-timing always takes priority over line-timing for system 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 that 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 having the following privileges:

- **Security Administrators:** have read-write privileges and can execute all CLI commands, including those available to Administrators, Operators, and Inspectors
- **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 execute all system commands, including those available to the Operators and Inspectors.
- **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.
- **Inspectors:** are limited to a few read-only Exec Mode commands. The bulk of these are **show** commands for viewing a variety of statistics and conditions. An Inspector cannot execute **show configuration** commands and does not have the privilege to enter the Config Mode.

Configuration 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 Packet Processing 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 <pwd> | encrypted password <pwd> }
  end
```

Notes:

- Additional keyword options are available that identify active administrators 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 <pwd> | encrypted password <pwd> }
  end
```

Notes:

- Additional keyword options are available that identify active administrators 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 <pwd> | encrypted password <pwd> }

end
```

Notes:

- Additional keyword options are available that identify active administrators 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 <pwd> | encrypted password <pwd> }

end
```

Notes:

- Additional keyword options are available that identify active administrators 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 sample output for this command. 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 identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* 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. A sample output for this command appears below. 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 TACACS+ for System Administrative Users

This section describes TACACS+ (Terminal Access Controller Access Control System+) AAA (Authentication Authorization and Accounting) service functionality and configuration on the ASR 5000.

Operation

TACACS+ is a secure, encrypted protocol. By remotely accessing TACACS+ servers that are provisioned with the administrative user account database, the ASR 5000 can provide TACACS+ AAA services for system administrative users. TACACS+ is an enhanced version of the TACACS protocol, and uses TCP instead of UDP.

The ASR 5000 system serves as the TACACS+ Network Access Server (NAS). As the NAS the system requests TACACS+ AAA services on behalf of authorized system administrative users. For the authentication to succeed, the TACACS+ server must be in the same local context and network accessed by the system.

The system supports TACACS+ multiple-connection mode. In multiple-connection mode, a separate and private TCP connection to the TACACS+ server is opened and maintained for each session. When the TACACS+ session ends, the connection to the server is terminated.

TACACS+ is a system-wide function on the ASR 5000. TACACS+ AAA service configuration is performed in TACACS Configuration Mode. Enabling the TACACS+ function is performed in the Global Configuration Mode. The system supports the configuration of up to three TACACS+ servers.

Once configured and enabled on the system, TACACS+ authentication is attempted first. By default, if TACACS+ authentication fails, the system then attempts to authenticate the user using non-TACACS+ AAA services, such as RADIUS.

User Account Requirements

Before configuring TACACS+ AAA services on the ASR 5000, note the following TACACS+ server and system user account provisioning requirements:

TACACS+ User Account Requirements

The TACACS+ server must be provisioned with the following TACACS+ user account information:

- A list of known administrative users.
- The plain-text or encrypted password for each user.

- The name of the group to which each user belongs.
- A list of user groups.
- TACACS+ privilege levels and commands that are allowed/denied for each group.



Important: TACACS+ privilege levels are stored as Attribute Value Pairs (AVPs) in the network's TACACS+ server database. Users are restricted to the set of commands associated with their privilege level. A mapping of TACACS+ privilege levels to ASR 5000 CLI administrative roles and responsibilities is provided in the table below.

Table 3. Mapping of TACACS+ Privilege Levels to CLI Administrative Roles

TACACS+ Privilege Level	CLI Administrative Access Privileges				
	CLI	FTP	ECSEMS	Lawful Intercept	CLI Role
0	Yes	No	No	No	Inspector
1	Yes	No	Yes	No	Inspector
2	No	Yes	No	No	Inspector
3	Yes	Yes	No	No	Inspector
4	Yes	Yes	Yes	No	Inspector
5	Yes	No	No	No	Operator
6	Yes	No	Yes	No	Operator
7	No	Yes	No	No	Operator
8	Yes	Yes	No	No	Operator
9	Yes	Yes	Yes	No	Operator
10	Yes	No	No	No	Administrator
11	Yes	No	Yes	No	Administrator
12	No	Yes	No	No	Administrator
13	Yes	Yes	No	Yes	Administrator
14	Yes	Yes	Yes	No	Administrator
15	Yes	Yes	Yes	Yes	Administrator

User Account Requirements

TACACS+ users who are allowed administrative access to the system must have the following user account information defined on the ASR 5000:

- username
- password
- administrative role and privileges



Important: For instructions on defining users and administrative privileges on the system, refer to the *Configuring System Administrative Users* chapter in this guide.

Configuring TACACS+ AAA Services

This section provides an example of how to configure TACACS+ AAA services for administrative users on the system.



Caution: When configuring TACACS+ AAA services for the first time, the administrative user must use non-TACACS+ services to log into the ASR 5000. Failure to do so will result in the TACACS+ user being denied access to the system.

Log in to the system using non-TACACS+ services.

Use the example below to configure TACACS+ AAA services on the system:

configure

tacacs mode

server priority <priority_number> **ip-address** <tacacs+srvr_ip_address>

end

Note:

- **server priority** *priority_number*: Must be a number from 1 to 3, that specifies the order in which this TACACS+ server will be tried for TACACS+ authentication. 1 is the highest priority, and 3 is the lowest.
- **ip-address**: Must be the IPv4 address of a valid TACACS+ server that will be used for authenticating administrative users accessing this system via TACACS+ AAA services.
- By default, the TACACS+ configuration will provide authentication, authorization, and accounting services.

Enable TACACS+ on the ASR 5000:

configure

aaa tacacs+

end

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



Important: For complete information on all TACACS+ Configuration Mode commands and options, refer to the *TACACS Configuration Mode Commands* chapter in the *Command Line Reference*.

Verifying the TACACS+ Configuration

This section describes how to verify the TACACS+ configuration:

Log out of the system CLI, then log back in using TACACS+ services.



Important: Once TACACS+ AAA services are configured and enabled on the ASR 5000, the system first will try to authenticate the administrative user via TACACS+ AAA services. By default, if TACACS+ authentication fails, the system then continues with authentication using non-TACACS+ AAA services.

At the Exec Mode prompt, enter the following command:

```
show tacacs
```

The command output provides summary information for each active TACACS+ session such as username, login time, login status, current session state and privilege level.

An example of this command's output is provided below. In this example, a system administrative user named *asradmin* has successfully logged in to the system via TACACS+ AAA services.

active session #1:

```
login username : asradmin
login tty      : /dev/pts/1
time of login  : Fri Oct 22 13:19:11 2011
login server priority : 1
current login status : pass
current session state : user login complete
current privilege level : 15
remote client application : ssh
remote client ip address : 111.11.11.11
last server reply status : -1
```

total TACACS+ sessions : 1



Important: For details on all TACACS+ maintenance commands, refer to the *show tacacs* chapter in the *Statistics and Counters Reference*.

Configuring a Chassis Key

A unique chassis key is configured at the factory for each system. This key is used to decrypt encrypted passwords found in generated configuration files. The system administrator can create a unique chassis key that will be used to encrypt passwords stored in configuration files.

The Exec mode **chassis key value** *key_string* command identifies the chassis which can encrypt and decrypt encrypted passwords in the configuration file. If two or more chassis are configured with the same chassis key value, the encrypted passwords can be decrypted by any of the chassis sharing the same chassis key value. As a corollary to this, a given chassis key value will not be able to decrypt passwords that were encrypted with a different chassis key value.

The *key_string* is an alphanumeric string of 1 through 16 characters. The chassis key is stored as a one-way encrypted value, much like a password. For this reason, the chassis key value is never displayed in plain-text form.

The Exec mode **chassis keycheck** *key_string* command generates a one-way encrypted key value based on the entered *key_string*. The generated encrypted key value is compared against the encrypted key value of the previously entered chassis key value. If the encrypted values match, the command succeeds and keycheck passes. If the comparison fails, a message is displayed indicating that the key check has failed. If the default chassis key (no chassis key) is currently being used, this key check will always fail since there will be no chassis key value to compare against.

Use the **chassis keycheck** command to verify whether multiple chassis share the same chassis key value.



Important: Only a user with Security Administrator or Administrator privilege can execute the **chassis key value** and **chassis keycheck** commands.

For additional information, refer to the *Exec Mode Commands (A-C)* chapter in the *Command Line Interface Reference*.

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 EPROM 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 line card 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 to 00:05:47:FF:FF:FF) for use by customers. This range allows you to create 256 address blocks each containing 256 MAC addresses (for example, 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 the *Understanding System Operation and Configuration* chapter, when the system boots up, all installed packet processing cards 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 packet processing cards 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 packet processing cards 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.

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 packet processing card or SMC.

Use the following example to configure packet processing card 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 packet processing card in the chassis that you wish to activate.
- **card-standby-priority** specifies the order in which the system will use standby packet processing cards as redundant components.
 - By default, the system uses the standby packet processing card 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 packet processing card 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 packet processing card that you want to use as the last redundant component.

Example

A system has three packet processing cards 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 packet processing card in slot 15 to replace the failed packet processing card followed by the packet processing card in slot 14, enter the following command:

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

In the unlikely event that the packet processing cards in chassis slots 15 and 14 are unavailable, the system automatically uses the remaining standby packet processing card 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 packet processing card 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 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 5. Network Topology Example Using Line Card Port Redundancy

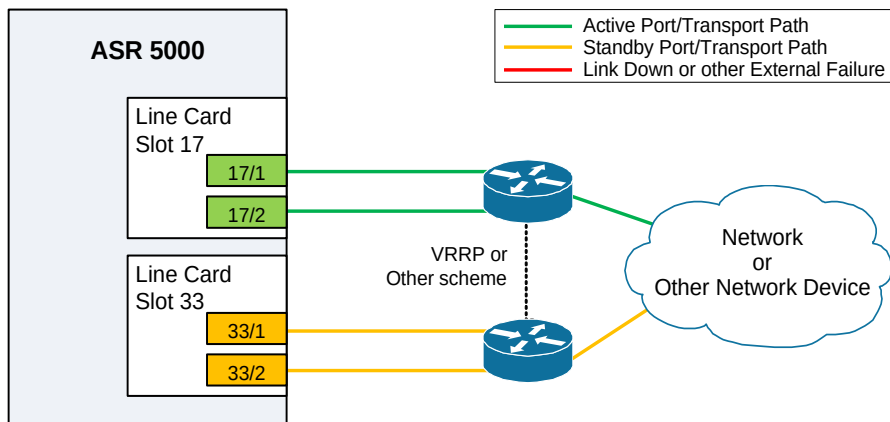
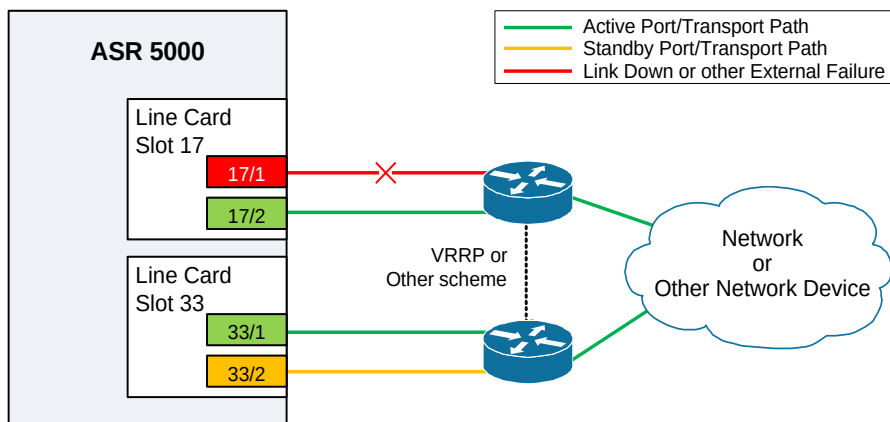
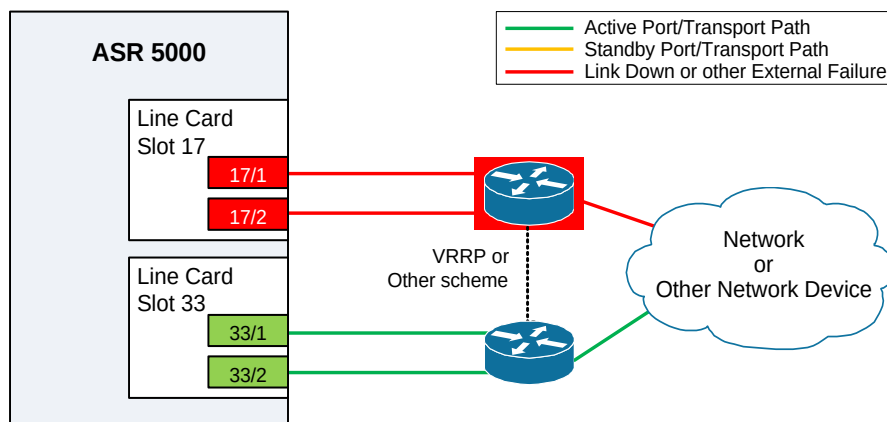


Figure 6. Port Redundancy Failover in Cable Defect Scenario



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 7. 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-SMC chassis with SPIO port redundancy enabled, you should have CFE1.1.0 or greater in flash on both SMCs. Otherwise, you risk having a standby 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 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
```

■ Configuring Line Card and SPIO Port Redundancy

Link Duplex : Unknown

Link Speed : Unknown

Logical ifIndex : 285278209

Operational State : Down, Active

Configuring ASR 5000 Link Aggregation

A Link Aggregation Group (LAG) works by exchanging control packets via Link Aggregation Control Protocol (LACP) Marker Protocol over configured physical ports with peers to reach agreement on an aggregation of links as defined in IEEE 802.3ad. The LAG sends and receives the control packets directly on physical ports attached to different XGLC (10 Gig Ethernet) or QGLC (Quad Ethernet 1000) cards.

Link aggregation (also called trunking or bonding) provides higher total bandwidth, auto-negotiation, and recovery by combining 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.

A LAG can be formed by configuring:

- Ports from the same QGLC
- Ports from top-bottom QGLCs with port redundancy
- Ports from side-by-side XGLCs
- Ports from horizontal QGLCs where ports could be from different cards, while some cards could be in L2 (side-by-side) redundancy.



Important: LAG cannot be configured when XGLCs have been configured for side-by-side redundancy. A failure message appears on the CLI when such an attempt is made.

LAG and Master Port

Logical port configurations (VLAN and binding) are defined in the master port of the LAG. If the master port is removed because of a card removal/failure, another member port becomes the master port (resulting in VPN binding change and outage), unless there is a redundant master port available.

LAG and Port Redundancy

A LAG supports redundant ports, either top-down on the QGLC (vertical link aggregation) or horizontal link aggregation on the XGLC, when only one port is active in the pair. By default, active ports in a LAG can be on any XGLC or QGLC, but optionally, all ports in a LAG can be auto-switched to another card when certain active port counts or bandwidth thresholds are crossed.

LAG and Multiple Switches

This feature connects ports on XGLCs or QGLCs to ports on Ethernet switches. A port failure/switch forces all ports in a LAG to switch to another XGLC or QGLC when a specified threshold is crossed. This works in a way similar to the auto-switch feature for port redundancy. LACP runs between the ASR 5000 and the Ethernet switch, exchanging relevant pieces of information, such as health status.

The following table summarizes the functionality of LAG with QGLC and XGLC cards.

Table 4. QGLC and XGLC LAG Functionality

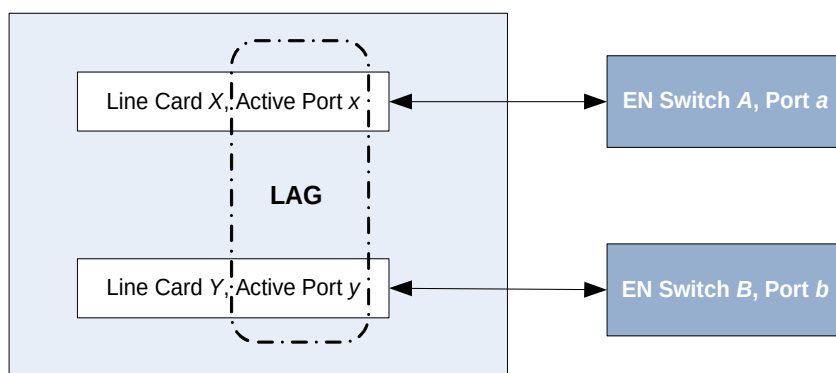
Card Type	ASR 5000	LAGID	Ethernet Switch A	Ethernet Switch B
QGLC	Port 1	1	Port 1	----
	Port 2	1	Port 2	----
	Port 3	1	----	Port 1
XGLC	Slot 1	1	Port 1	----
	Slot 2	1	----	Port 1

Multiple Switches without L2 Redundancy

If one LAG is connected to different peers, by default, the implementation selects the higher bandwidth peer to form aggregation. If there are several horizontal cards with LAG ports that are all in active mode (no L2 redundancy) connected to different switches, each card provides a candidate aggregation of bandwidth. Selection by bandwidth works because the failure of one port causes that card's bandwidth to be lower, thus causing another card to be selected.

The figure below shows an LAG established across two line card ports without L2 redundancy.

Figure 8. LAG without L2 Redundancy, Two Ethernet Switches



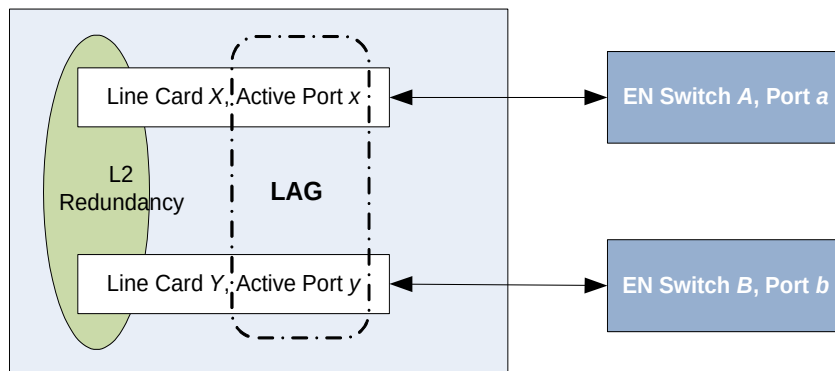
Multiple Switches with L2 Redundancy or Active-Active Mode

To handle the implementation of Link Aggregation Control Protocol without requiring standby ports to pass LACP packets, two separate instances of LACP are started on redundant cards. The two LACP instances and port link state are monitored to determine whether to initiate an auto-switch (including automatic L2 port switch).

Two switches can also be connected to odd and even slots of an XGLC in active-active mode without L2 redundancy. Two LACP instances are started for odd and even slots, and similar monitoring and switching occurs.

The figure below shows an LAG established across two line card ports with L2 redundancy.

Figure 9. LAG with L2 Redundancy, Two Ethernet Switches



An LACP implementation with L2 redundancy cannot pass traffic even though standby ports have link up. For example, with two QGLC cards connected to two different Ethernet switches configured as a top-bottom pair and all ports in the same LAG, failure of ports would not trigger a LAG switch until the active port number ratio flipped (more ports down than up).

To handle this case without requiring standby ports to pass LACP packets, separate instances of LACP are started on redundant cards. The LAG manager monitors the two LACP instances and port link states to decide whether to auto-switch (including automatic L2 port switching).

Port States for Auto-Switch

Ports are classified in one of the four states (shown in the following table) to determine whether to start auto-switching. For counters, State(x) represents the number of ports on a card in that state.

Table 5. Auto-Switch Port States

State	Counter	Description
Link	L(x)	Physical link up
Standby	S(x)	Link up but in standby mode
Waiting	W(x)	Waiting for Link Aggregation Control Protocol negotiation
Aggregated	A(x)	Aggregation formed

Hold Time

Once the LAG manager switches to another LACP instance, it does not consider another change for a short period to let link and LACP negotiation settle down. This “hold time” is configurable.

The LAG manager also enters/extends the hold period when an administrator manually switches ports to trigger a card switch.

Preference and Revertive Mode

You can define which card is preferred to implement revertive mode. This preference is defined per LAG group. Port preference is not allowed in this mode.

Auto-Switch Criteria

The following criteria determine the switching of card x to card y to provide better bandwidth while allowing manual intervention. The evaluation of the criteria occurs outside of the hold period.

Ports are automatically switched from card x to card y when $A(y) \neq 1$, at least one port is in aggregated state on card y , and one of the following conditions is true (in order of precedence):

- $L(x) > L(y)$ Less ports with link Up on card x than card y
- $S(x) > S(y)$ More ports in Standby state on card x than card y
- $W(x) > W(y)$ More ports in Waiting state on card x than card y
- $A(x) < A(y)$ Fewer ports in Aggregated state on card x than card y
- Card y is preferred
- Card y is selected.

Distribution Options

The specified distribution option controls how the LAG hash map is generated. This option is set on the master port for use by the whole LAG via the **link-aggregation distribution** command. The following table identifies the distribution options (assuming port index 0,1,2,3 were selected):

Table 6. LAG Distribution Options

Distribution	Meaning	Example
Simple	Repetition of all selected port indices	0123012301230123...
Rotate	Repetition of the rotated port indices	0123123023013012...
Block	Blocks of the same port index	0000111122223333
Random	Based on pseudo random number	—

QGLC Link Aggregation

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. Top and bottom QGLC cards can be connected to different switches in a LAG.

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.
- 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.
- Certain physical port configuration changes, such as the MAC address or SRP, are prohibited on any interface participating in link aggregation

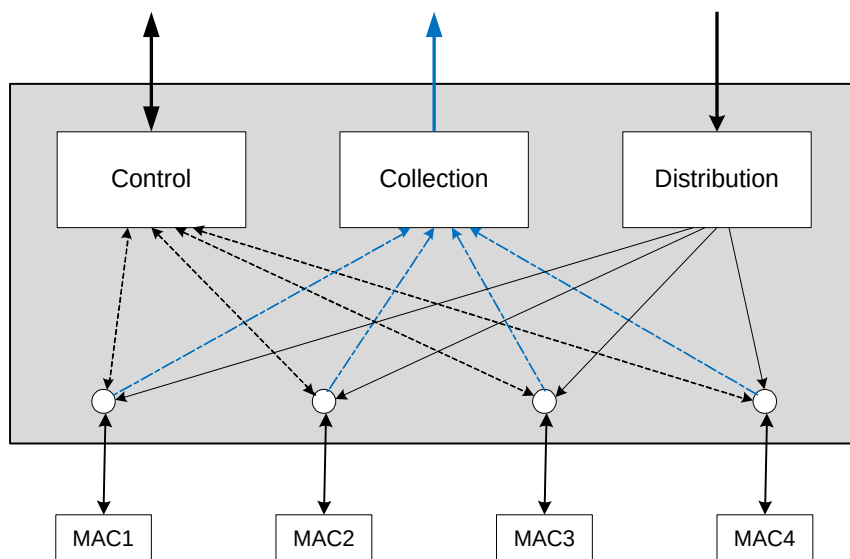
There is more on configuring ports and port redundancy in *Configuring Line Card and SPIO Port Redundancy*.

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 10. QGLC 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.

XGLC Link Aggregation

Because the XGLC is a full-height card that requires top and bottom card slots, link aggregation takes place horizontally within ports on different XGLCs.

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 packet services card 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 or XGLC 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.

Redundancy Options

For redundancy, there is an option that controls the auto-switching of L2 redundant or active-active ports when they are connected to two switches. Set this option on the master port for use with the whole group

```
link aggregation [redundancy { link aggregation toggle link | switched }] [hold-time sec]
[ preferred slot {card_number | none }
```

Distribution Options

This option controls how a LAG hash map is generated. This method is required in the case of ECMP over LAG. Set this option on the master port for use with the whole group.

```
link aggregation { simple | rotate | block | random }
```

The following list defines the distribution options (assuming port index 0,1,2,3 were selected).

- **simple**: Repetition of all selected port indexes (Example: 0123012301230123...)
- **rotate**: Repetition of rotated port index (Example: 0123123023013012...)
- **block**: Blocks of the same port index (Example: 0000111122223333)
- **random**: Based on pseudo random number

Horizontal Link Aggregation with Two Ethernet Switches

When a LAG contains two sets of ports each connecting to a different switch, the operator has the ability to specify the slot/port (connected to the destination switch) when switching ports.

The Exec mode **link-aggregation port switch to slot/port** command is used to configure this option. The following criteria apply to the setting of this option:

- *slot/port* must support LAG.
- *slot/port* must be configured with LAG.
- *slot/port* must not be already actively distributing
- *slot/port* must have negotiated a link aggregation partner in standard mode.
- *slot/port*'s partner must have an equal or higher in standard mode.
- *slot/port*'s partner bundle must have equal or higher bandwidth in standard mode.
- Switching to *slot/port* must not violate preference within hold-time in standard mode.

Link Aggregation Status

To check the status of link aggregation, use the following commands:

- **show port table**
- **show port info slot/port**

Chapter 4

Configuring Management Settings

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

This chapter includes the following sections:

- [ORBEM and the Web Element Manager](#)
- [Configuring ORBEM Client and Port Parameters](#)
- [Configuring Internet Inter-ORB Protocol \(IIOP\) Transport Parameters](#)
- [Verifying ORBEM Parameters](#)
- [SNMP Support](#)
- [Controlling SNMP Trap Generation](#)

ORBEM and the Web Element Manager

The system can be managed by a Common Object Broker Request Architecture (CORBA)-based, element management system application called the Cisco® Web Element Manager (WEM).

You must configure the ORBEM settings that allow the system to communicate with the server running the WEM application.



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 configuration 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 ORBEM 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 via the **event-notif-service** command, you can set 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 WEM default process enforces SSL.

Verifying ORBEM Parameters

- Step 1** Run the **show orbem client table** command to verify that the client was configured properly. This command lists the configured ORBEM clients and displays their state and privileges.
- Step 2** Run the **show orbem status** command to verify the ORBEM parameter configuration. 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
```

SNMP Support

The system uses the SNMP to send traps or events to the Web Element Manager server or an alarm server on the network. You must configure SNMP settings to communicate with those devices.



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.

The *SNMP MIB Reference* describes the MIBs and SNMP traps supported by the ASR 5000 platform. A list of unsupported MIBs and traps appears in the *Unsupported Management* appendix to this guide.

Configure the system to communicate with the WEM server or an alarm server:

- Step 1** Set 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, WEM can only process SNMP version 1 (SNMPv1) and SNMP version 2c (SNMPv2c) traps. If the SNMP target you are configuring is the WEM 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 entities 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 by the WEM application.

Verifying SNMP Parameters

- Step 1** Run the **show snmp server** command to verify that the SNMP server information is correctly configured. The following displays a sample output of this command.

```
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. 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.

```
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 5

Verifying and Saving Your Configuration

This chapter describes how to save your system configuration and includes the following sections:

- [Verifying the Configuration](#)
- [Saving the Configuration on the ASR 5000 Chassis](#)

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. An example includes IP address pool configuration. Using this example, enter the following commands to verify proper feature configuration:

Enter the **show ip pool** command to display the IP address pool configuration. 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 *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 : egtpl
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 **show context name name** command.

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 **show configuration** command.

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 **show configuration errors** command.

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 as shown in the examples below.

```
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 on the ASR 5000 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] [obsolete-encryption]
[showsecrets] [verbose]
```

Table 7. Command Syntax for Saving the Configuration

Keyword/Variable	Description
<i>url</i>	Specifies the path and name to which the configuration file is to be stored. <i>url</i> may refer to a local or a remote file. <i>url</i> must be entered using one of the following formats: <pre>file: { /flash /pcmcia1 hd-raid } [/directory] /filename> tftp:// { ipaddress host_name [:port#] } [/directory] /filename ftp:// [username [:pwd] @] { ipaddress host_name } [:port#] [/directory] /filename sftp:// [username [:pwd] @] { ipaddress host_name } [:port#] [/directory] /filename</pre> /flash corresponds to the CompactFlash on the SMC. /pcmcia1 corresponds to PCMCIA slot 1. /hd-raid corresponds to the RAID hard disk drive array on the SMC. <i>ipaddress</i> is the IP address of the network server. <i>host_name</i> is the network server's <i>hostname</i>. <i>port#</i> is the network server's logical port number. Defaults are: - tftp: 69 - data - ftp: 20 - data, 21 - control - sftp: 115 - data Note: <i>host_name</i> can only be used if the networkconfig parameter is configured for DHCP and the DHCP server returns a valid <i>nameserver</i>. <i>username</i> is the username required to gain access to the server if necessary. <i>password</i> 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	<i>Optional:</i> This keyword directs the system to save the CLI configuration file to the local device, defined by the <i>url</i> 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 similar 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: When saving the file to an external network (non-local) device, the system disregards this keyword.
-noconfirm	<i>Optional:</i> 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).
obsolete-encryption	Saves configuration with encrypted values generated from an obsolete encryption method.

Keyword/Variable	Description
showsecrets	<i>Optional:</i> Saves the CLI configuration file with all passwords in plain text, rather than their default encrypted format.
verbose	<i>Optional:</i> Displays every parameter that is being saved to the new configuration file.



Important: Do not use the “/” (forward slash), “:” (colon) or “@” (at sign) characters when entering a string for the following URL fields: directory, filename, username, password, host or port#.



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 CompactFlash in the SMC, 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 6

System Interface and Port Configuration Procedures

This chapter describes how to create a context and configure system interfaces and ports within the context. Before beginning these procedures, refer to your product-specific administration guide for configuration information for your product.

This chapter includes the following:

- [Creating Contexts](#)
- [Creating and Configuring Ethernet Interfaces and Ports](#)
- [Creating and Configuring ATM Interfaces and Ports](#)
- [Creating and Configuring Frame Relay Interfaces and Ports](#)



Important: Make sure at least one packet processing card is active before you configure system elements. Refer to *Configuring System Settings* in this guide for information and instructions on activating packet processing cards.

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 represent the most common or likely commands and/or keyword options. In many cases, other 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
    
```

Repeat to configure additional contexts.



Important: We recommend that if your system is using Fast Ether Line Cards (FELCs, Ethernet 10/100), 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 similar to the example below. This example shows that two contexts were created: one named *source* and one named *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 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](#) section.
- 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 each interface to be configured.



Important: This section provides the minimum instructions 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 Commands* and *Ethernet Interface Configuration Mode Commands* 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 | ipv6 } address <address subnetmask> [ secondary ]

  end
```

Notes:

- *Optional:* Add the **loopback** keyword option to the **interface name** command, to set the interface type as “loopback” which is always UP and not bound to any physical port.
- *Optional:* Add the **secondary** keyword to the { **ip** | **ipv6** } **address** command, to assign multiple IP addresses to the interface. IP addresses can be entered using IPv4 dotted-decimal or IPv6 colon-separated notation.
- *Optional:* 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 IP address can be entered using IPv4 dotted-decimal or IPv6 colon-separated notation.

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.
- For **port ethernet port#**, on FELCs, this value can be from 1 to 8. For the GELC/GLC2s and XGLCs this value must be 1. For for QGLCs enter a value from 1 to 4.
- *Optional:* In the Ethernet Port configuration mode, add the preferred **slot slot#** command if line card port redundancy was enabled at the card level and you want to specify a port preference.
- *Optional:* 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 named interface.

Configuring a Static Route for an Interface

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

```
configure

  context <name>

    { ip | ipv6 } 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. This IP address can be entered using IPv4 dotted-decimal or IPv6 colon-separated notation.
- *gw_address* is the IP address of the default gateway or next-hop route. This IP address can be entered using IPv4 dotted-decimal or IPv6 colon-separated notation.
- 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 | ipv6 } interface
```

context_name represents the name of the context in which the interface was created. The output from these commands should be similar to 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 State:           UP (Bound to 17/1 untagged, ifIndex 285278209)

IP State:           UP (Bound to 5/11 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.

slot# is the chassis slot number of the MIO on which the physical port resides. *slot#* can be 5 or 6.

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.

This command produces an output similar to that displayed in the following example that shows the configuration for port 1 on the virtual line card installed in 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	SPI01
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 (IPoA) through an Optical ATM line card (OLC/OLC2). The procedures describe how to:

- 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](#) section.
- 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 an attached remote) to ensure transmit synchronization by applying the example configuration in the *Configuring Transmit Timing Source* section of the *Configuring System Settings* chapter.
- Step 6** Verify the port and interface configuration with the procedure [Verifying Port and Interface Configuration](#) section.
- 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 and set the framing type:

```
configure

card <slot#>

    framing <SDH|SONET>

    no shutdown

end
```

Notes:

- The default framing type is SONET (Synchronous Optical Network) 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 (Signalling System No. 7) 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 Commands* 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/CLC2) 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 identifiers (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 the verification examples below.
- 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** (24 timeslots, 1.536 Mbps) or **e1** (32 timeslots, 2.048 Mbps), 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/CLC2 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 record 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 data link connection identifier (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

```

■ Creating and Configuring Frame Relay Interfaces and Ports

```

        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 of the **show** command 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 7

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](#)
- [Maintaining the Local File System](#)
- [Configuring the Boot Stack](#)
- [Upgrading the Operating System Software](#)
- [Managing License Keys](#)
- [Managing Local-User Administrative Accounts](#)

Understanding the Local File System

The 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 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 on the front panel of the SMC.
- **/hd-raid** - This 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 via the 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.
- **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.
- **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 system 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 the *Getting Started* chapter 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 | a ll } [checkonly] [reverse] } [-
noconfirm ]
```

Table 8. Command Syntax Descriptions

Keyword/Variable	Description
<code>/flash</code>	Synchronizes only the CompactFlash file system on the standby SPC/SMC.
<code>/pcmcia1</code>	Synchronizes only the file system of the PCMCIA card installed in the PCMCIA 1 slot on the standby SPC/SMC.
<code>all</code>	Specifies that filesystems on all available matching local devices (be synchronized.
<code>checkonly</code>	Displays a list of files that can be synchronized without executing any synchronization actions.
<code>reverse</code>	Performs the specified operation on the standby SPC/SMC.
<code>-noconfirm</code>	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 SMC /flash devices.

```
card smc 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 | /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.

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

Table 9. rename command options

Keyword/Variable	Description
<i>src_filename</i>	The name of the source file, with its extension, that you are renaming.
<i>dst_filename</i>	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.
-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 *iot_test.cfg* to *iot_accept.cfg* on the */flash* local device.

```
rename /flash/iot_test.cfg /flash/iot_accept.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 on the ASR 5000 Chassis

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

```
[local]host_name#
```

To copy files, enter the following command:

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

Table 10. Command Syntax for Copying a File

Keyword/Variable	Description
<i>from_url</i> <i>to_url</i>	Specifies the path name from which the configuration file is to be copied and the one to which it will be written. <i>url</i> may refer to a local or a remote file. <i>url</i> must be entered using one of the following formats: file: { /flash /pcmcia1 hd-raid } [/directory] /<filename> tftp:// { <i>ipaddress</i> <i>host_name</i> [:port#] } [/directory] /filename ftp:// [<i>username</i> [:pwd] @] { <i>ipaddress</i> <i>host_name</i> } [:port#] [/directory] /filename sftp:// [<i>username</i> [:pwd] @] { <i>ipaddress</i> <i>host_name</i> } [:port#] [/directory] /filename /flash corresponds to the CompactFlash on the SMC. /pcmcia1 corresponds to PCMCIA slot 1. /hd-raid corresponds to the RAID hard disk drive array on the SMC. <i>ipaddress</i> is the IP address of the network server. <i>host_name</i> is the network server's <i>hostname</i>. <i>port#</i> is the network server's logical port number. Defaults are: - tftp: 69 - data - ftp: 20 - data, 21 - control - sftp: 115 - data Note: <i>host_name</i> can only be used if the networkconfig parameter is configured for DHCP and the DHCP server returns a valid <i>nameserver</i>. <i>username</i> is the username required to gain access to the server if necessary. <i>password</i> 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.
-noconfirm	<i>Optional:</i> 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).
passive	Use passive mode FTP.

To copy a configuration file called *system.cfg* from a directory that was called *cfgfiles* to a directory named *configs_old* on the CompactFlash in the SMC, enter the following command:

```
copy /flash/cfgfiles/system.cfg /flash/configs_old/system_2011.cfg
```

To copy a configuration file called *simple_ip.cfg* from a directory called *host_name_configs* to 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:

```
copy /flash/host_name_configs/simple_ip.cfg
ftp://administrator:secure@192.168.34.156/host_name_configs/
simple_ip.cfg
```

To copy 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:

```
copy /flash/cfgfiles/init_config.cfg tftp://config_server/init_config.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.

```
delete { /flash | /pcmcial | /hd-raid } /filename [-noconfirm]
```

Table 11. delete command variables

Keyword/Variable	Description
<i>filename</i>	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 `/flash` directory.

```
delete /flash/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. rmdir command options

Keyword/Variable	Description
<i>url</i>	Specifies the path name of the local device. The <i>url</i> variable must be formatted as follows: <code>{/flash /pcmcial /hd-raid}/<directory>/<file_name></code> <code>file:{/flash /pcmcial /hd-raid}/<directory>/<file_name></code>
<i>dir_name</i>	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* in the */flash* directory.

```
rmkdir /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, when placing a new local device into the SMC for regular use, you should format the device via 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: The **format** command removes all files and information stored on the device.

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

```
format { /flash | /pcmcial | /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 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
<i>url</i>	Specifies the location of the CLI configuration file to be applied. The <i>url</i> may refer to a local or a remote file. The URL must be formatted according to one of the following formats: ASR 5000: • <code>{/flash /pcmcial /hd-raid}/<directory>/<file_name></code> • <code>file:/{/flash /pcmcial /hd-raid}/<directory>/<file_name></code> • <code>tftp://{<ipaddress> <host_name>}/<directory>/<file_name></code> • <code>ftp://[<username>[:<password>]@]<ipaddress> <host_name>}/<directory>/<file_name></code> • <code>sftp://[<username>[:<password>]@]<host>[/<path>]/<filename></code> • <code>http://[<username>[:<password>]@]<host>[/<path>]/<filename></code> Where <i>username</i> is the user to be authenticated as, <i>password</i> is the password to use for authentication, <i>host</i> is the IP address using the standard IPv4 or the logical host name, and <i>path</i> is the directory structure to the file. <i>filename</i> is the name of the CLI configuration file or script that you wish to apply. The use of long filenames is supported.
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* in the */flash* directory.

```
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:

```
directory { /flash | /pcmcial | /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:

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

Where: *url* is the path name for the location of the file and *filename* is the name of the file, including any extension.



Important: Operator and inspector-level users can execute the **show file** command but cannot 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 (for example, the file was truncated or was transferred using ASCII mode instead of binary mode), 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:

```
show version { /flash | /pcmcia1 | /hd-raid } /[directory]/filename [all]
```

The output of this command displays the following information:

- Version – NN.N (build number)
- Description
- Date
- Boot Image
- Size

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 (for example, 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 local-boot method 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 or **/hd-raid** located on the active 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.

The system can also 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 run the Exec mode **show boot** command.



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 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.st40.bin
\config /flash/general_config.cfg

boot system priority 19 image tftp://192.168.1.161/tftpboot/build14489.st40.bin
\config /flash/general_config.cfg

boot system priority 20 image /flash/build14456.st40.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>config
```

Table 14. boot system priority command options

Keyword/Variable	Description
<i>number</i>	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. An existing priority number, used by another boot stack entry, may be entered. However, this will overwrite the existing entry in the boot.sys file.
<i>image_url</i>	Specifies the location of the operating system software image file to be applied. The URL may refer to a local or a remote device. It must be formatted as follows: [file:] {/flash pcmcial hd-raid} [/<path>] /<file_name> [http: tftp:] //<host>[:<port>] [/<path>] /<filename>
<i>cfg_url</i>	Specifies the name of the local device and directory path where the CLI configuration file is located. The URL must be formatted as follows: [file:] {/flash pcmcial hd-raid} [/<path>] /<file_name> Where <i>path</i> is the directory structure to the file, and <i>file_name</i> is the name of the configuration file to load. This file typically has a .cfg extension.

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 rebooted.

Synchronize the local file systems on the SMCs by the following command:

```
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 management LAN interface that the system will use to communicate with the management network when using the network booting method.

**Important:** Booting a dual-SMC chassis with SPIO port redundancy enabled requires that both SMCs have CFE1.1.0 or greater in flash. If CFE1.1.0 or greater is not present on both cards, the standby SMC may not be able to boot from the network in certain circumstances.

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 { local-eth1 | local-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. local-eth1 corresponds to the port 1 interface on the SPIO. local-eth2 interface that corresponds to the port 2 interface on the SPIO. NOTE: Use port 1 for network booting.

Keyword/Variable	Description
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 (10), 100 Mbps (100), or 1000 Mbps (1000). This command keyword must be following by the speed of the Ethernet connection, entered as an integer. NOTE: 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. NOTE: Ethernet networking rules dictate that if a device's interface is configured for auto-negotiation is communicating with a device that is manually configured to support full duplex, the first device will negotiate to the manually configured speed of the second device, but will only communicate in half duplex mode.
media	The media for MIO port 1 is fixed at rj45 . 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. NOTE: If this option is selected, you will not have to configure IP address information for the SPIO interface, defined using the boot interface command, or any needed gateway.
dhcp-static-fallback	Specifies the use of the DHCP to automatically assign an IP address to the SPIO interface, defined using the boot interface command, at startup. However, this option allows the configuration of a fallback static IP address that can be used when the DHCP server is unreachable.

Keyword/Variable	Description
static	Specifies that a static IP address will be configured for the SPIO 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 <i>ip_address24</i>	Specifies the IP address to use for the SPIO interface in slot 24. Enter the <i>ip_address24</i> variable as an IP address in IPv4 dotted-decimal notation.
spio25 <i>ip_address25</i>	Specifies the IP address to use for the SPIO interface in slot 25. Enter the <i>ip_address25</i> variable as an IP address in IPv4 dotted-decimal notation. If used, both interfaces will appear in the boot.sys file.
netmask	Enter the subnet mask, using IPv4 dotted-decimal notation.
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 in IPv4 dotted-decimal notation 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 an integer 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 of the DNS server entered in IPv4 dotted-decimal notation.



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 commands issued 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 in the Exec mode by entering the **show version** command.

Saving a Copy of the Current Configuration File

Prior to upgrading to a new software release, you should copy and rename the current configuration file to the /flash drive and to an off-chassis location (external memory device or network URL). This renamed copy assures that you will have a fallback, loadable configuration file should a problem be encountered during the upgrade.

 **Important:** Configuration files created and saved in StarOS release 12.2 and higher cannot be shared across multiple chassis due to a change in the encryption algorithm for passwords and secrets. These release 12.2 + changes modify encrypted data in the configuration file so that it cannot be recognized by software builds prior to 12.2. If it is necessary to revert to a previous build, the chassis must be booted with the copy of the original configuration file. If this copy is not available, then the chassis must be loaded as if it is a new chassis.

Preparing for a Software Downgrade

If a software upgrade to release 12.2+ fails, you may need to downgrade to a previous software release, such as 12.0. Since the encryption algorithm for passwords and secrets is difference from 12.0 to 12.2+, you must save the 12.0 configuration file using the **obsolete-encryption** keyword to preserve passwords and secrets that used the earlier algorithm.

 **Important:** You must save the configuration prior to upgrading to release 12.2+.

The general sequence for a software downgrade in this scenerio is as follows:

- Develop a downgrade plan with the assistance of Cisco support personnel.
- Save the pre-12.2 configuration using the **obsolete-encryption** keyword in conjunction with the Exec mode **save configuration** command as shown in the example.


```
save configuration /flash/v120_system.cfg obsolete-encryption
```

Warning: Use of weaker encryption significantly reduces the security of the system

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

- To reboot the system using the old configuration, change the boot configuration via the Global Configuration mode to use the old boot image and its previously saved configuration. See the example below.

```
boot system priority 1 image /flash/v120_asr5000.bin config
/flash/v120_system.cfg
```

- Reload the system using the pre-12.2 image and configuration.
- Once the system has reloaded, verify that the configuration is correct.
- Save the current configuration as a backup once the configuration has been validated.

Software Upgrade Methods

Occasional software upgrades are required to add features and/or functionality, and to correct any previous defects. 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.

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 packet processing cards 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.

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 SMCs (one Active and one Standby)
- Two RCCs (required to support packet processing card migrations)
- Three packet processing cards (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.

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 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

The system remains in this Stage 1 until either all current sessions are self-terminated by users or the configured session upgrade limits are reached. In the latter 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 number of sessions) on the system is met and all sessions are terminated.

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.



Important: This is the only stage in which 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 packet processing cards to ensure that any errors are detected and that the re-directors used by the defined overload policy for each service remain in effect.

At this point, the SMCs are fully operational, but each packet processing card 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
- 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 packet processing cards 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 SMC switchover, wherein all tasks running on the active SMC are transferred to the standby SMC, which then becomes active and takes control of the system.

The new standby SMC is then restarted and the new operating system software image is loaded onto that 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 SMC is operational, another SMC switchover occurs and the second SMC is restarted, loading the new software version. During this period, since both 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 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 SMC is aware of all system and card-level states and tasks. All packet processing cards that are in standby operational mode are restarted simultaneously, and after passing their Power On Self Test (POST) diagnostics, their control processors (CPs) are loaded with the new operating system software image.

The remaining packet processing cards, which may be enforcing overload policies that prevent 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 packet processing cards. The original active packet processing cards are then restarted, all at once, and upgraded to the new operating system software image.



Important: The system will only migrate as many active packet processing cards as there are standby cards. If this is not a 1:1 correlation, the system will repeat this procedure of migrating - updating - migrating back until all normally active packet processing cards 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 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, depending on the size and complexity 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.

Performing an On-line Software Upgrade

This procedure describes how to perform a software upgrade 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.

Optional for PDSN: If you want to use the IP Pool Sharing Protocol during your upgrade, refer to the *Configuring IPSP Before the Software Upgrade* section of the *IP Pool Sharing Protocol* appendix in this administration guide.

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:

```
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 2012 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 2012 tworpcontext.cfg
-rwxrwxr-x 1 root root 15534 Aug 4 2012 test_vlan.cfg
-rwxrwxr-x 1 root root 2482 Nov 18 2011 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 (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 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: When transferring an operating system software image file via file transfer protocol (FTP), the FTP client must be configured to transfer the file using binary mode. Failure to use binary transfer mode will render 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 SMCs by entering one of the following commands:

```
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 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. <i>address</i> : 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.

Keyword/Variable	Description
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. <i>weight_num</i> 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	<i>Optional:</i> This 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 9 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 SMCs may result in unexpected behavior by the system that requires manual intervention to correct.

Step 10 Save your configuration as described in *Verifying and Saving Your Configuration*.

Step 11 Begin the on-line software upgrade by entering the following command:

```
upgrade online image_url config cfg_url [-noconfirm]
```

Keyword/Variable	Description
<i>image_url</i>	Specifies the location of the operating system software image file to be applied. The <i>url</i> may refer to a local or a remote device. The URL must be formatted as follows: [file:] {/flash /pcmcial /hd-raid} [/<path>] /<file_name> [http: tftp:] //<host>[:<port>] [/<path>] /<filename> Where <i>host</i> is an IP address or a logical host name, <i>path</i> is the directory structure to the file, and <i>filename</i> is the name of the operating system software image file. This file typically has a .bin extension. When using the TFTP, it is advisable to use a server that supports large blocks, per RFC 2348. This can be implemented by using the “block size option” to ensure that the TFTP service does not restrict the file size of the transfer to 32MB.

Keyword/Variable	Description
<code>cfg_url</code>	Specifies the name of the local device and directory path where the CLI configuration file is located. The URL must be formatted according as follows: <code>[file:]{/flash /pcmcia1 /hd-raid}/{/ <path> } / <file_name></code> Where <i>path</i> is the directory structure to the file, and <i>file_name</i> is the name of the configuration file to load. This file typically has a .cfg extension.
<code>-noconfirm</code>	Disables the “ Are you sure? [Yes No] ” confirmation prompt asked before executing the command.

The SMCs within the system load the new operating system image and the local file system is synchronized. The system then updates all standby packet processing cards. Next, it begins to update each active packet processing card, one at a time. The system monitors all sessions being processed by active packet processing cards. 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 packet processing cards in active mode to standby mode. Each new standby packet processing card is upgraded and rebooted. Once booted, the card is placed back into service as an active packet processing card.

Step 12 *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 SMCs 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.

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.

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:

```
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 2012 tworpcontext.cfg
-rwxrwxr-x 1 root root 15534 Aug 4 2012 test_vlan.cfg
-rwxrwxr-x 1 root root 2482 Nov 18 2011 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 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 SMCs by entering one of the following commands:

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

Keyword/Variable	Description
/flash	Synchronizes only the CompactFlash file system on the standby SMC.
/pcmcia1	Synchronizes only the file system of the PCMCIA card installed in the PCMCIA 1 slot on the standby SMC.
all	Specifies that filesystems on all available matching local devices (be synchronized. NOTE: Only filesystems on matching local devices will be synchronized. For example, if the active SMC contains two local devices (/flash and /pcmcia1) and the standby SMC contains only one local device (/flash), then synchronization would only occur on the matching local device.
checkonly	Displays a list of files that would be synchronized, without executing any synchronization actions.

Keyword/Variable	Description
reverse	Performs the specified operation on the standby SMC.
-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 as per your service requirements. 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 { asngw-service | asnpc-service | sgsn-service } { all | name
service_name } reject

newcall policy cscf-service { all | name service_name } { redirect
target_ip_address [ weight weight_num ] [ target_ipaddress2 [ weight weight_num ]
... target_ip_address16 [ weight weight_num ] | reject }

newcall policy { fa-service | lns-service | mip6ha-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}

newcall policy ggsn-service {apn name apn_name | all | name service_name}reject

newcall policy hnbgw-service {all | name service_name}reject

newcall policy {pcc-af-service | pcc-policy-service} {all | name
service_name}reject

newcall policy mme-service {all | name service_name } reject
```

Keyword/Variable	Description
name	Specifies a single instance of a service type or an APN to apply the newcall policy to. <i>service_name</i> is the name of a service that was previously configured. It can consist of up to 63 alphanumeric characters and is case sensitive. <i>apn_name</i> is the name of a previously configured APN expressed as an alphanumeric string of 1 through 63 characters that is case sensitive. 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. <i>address</i> : The IP address of an alternate destination expressed in IPv4 dotted-decimal format. 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.

Keyword/Variable	Description
weight <i>weight_num</i>	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. <i>weight_num</i> 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 alphanumeric characters. Note that *banner_text* must begin with and end in quotation marks (“”). For more information in entering CLI banner information, see the *CLI Reference*. 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 **show version** 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, refer to the *Configuring IPSP Before the Software Upgrade* section of the *IP Pool Sharing Protocol* appendix in this administration 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 default set of restricted 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.

SMCs are shipped with a CompactFlash card installed. A single license key is generated using the serial numbers from the CompactFlash cards. If two 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 SMCs, ensuring that licensed capacity and features remain available during a switchover event.

Session Use and Feature Use Licenses

Session use and feature use licenses are software mechanisms that provide session limit controls and enable special features within the system. These electronic licenses are stored in the system's configuration file that is loaded as part of the system software each time the system is powered on or restarted.

- Session use licenses limit the number of concurrent sessions that a system is capable of supporting per service type and are acquired on an as-needed basis. This allows carriers to pay only for what they are using and easily increase capacity as their subscriber base grows.
- Feature use licenses enable specific features/functionality within the system and are distributed based on the total number of sessions supported by the system.

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 license 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 an alphanumeric string of 1 through 1023 characters that is case sensitive. Copy the license key as shown in the example below, including the “\ (double-quote slash). 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|000000000
0000|LSP=000000|LSH=000000|LSG=500000|LSL=500000\FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|F
MG=Y|FCR=Y|FSR=Y|FPM=Y|FID=Y|SIG=MCwCF\Esnq6Bs/XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd638
8jHsHD99sg36SG267gshssja77
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 “\ (double-quote slash). 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|0000000000
0000|LSP=000000|LSH=000000|LSG=500000|LSI=500000\|FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|F
MG=Y|FCR=Y|FSR=Y|FPM=Y|FID=Y|SIG=MCwCF\Esnq6Bs/XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd638
8jHsHD99sg36SG267gshssja77

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 Cisco account 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 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 SMC is replaced, the CompactFlash card on the new SMC must be exchanged with the CompactFlash from the original SMC because the license key was generated based on the serial number of the CompactFlash card associated with the original 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 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 SMCs can be found in the *ASR 5000 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 in flash memory 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 8

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*.

This chapter includes the following sections:

- [SNMP Notifications](#)
- [Monitoring System Status and Performance](#)
- [Monitoring ASR 5000 Hardware Status](#)

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	<code>show administrators</code>
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	<code>show administrators session id</code>
View information pertaining to local-user administrative accounts configured for the system	<code>show local-user verbose</code>
View statistics for local-user administrative accounts	<code>show local-user statistics verbose</code>
View information pertaining to your CLI session	<code>show cli</code>
Determining System Uptime	
View system uptime (time since last reboot)	<code>show system uptime</code>
View NTP Server Status	
View NTP servers status	<code>show ntp status</code>
View System Resources	
View all system resources such as CPU resources and number of managers created	<code>show resources [cpu]</code>
View System Alarms	
View information about all currently outstanding alarms	<code>show alarm outstanding all verbose</code>
View system alarm statistics	<code>show alarm statistics</code>
View Congestion-Control Statistics	
View Congestion-Control Statistics	<code>show congestion-control statistics</code>
View Remote Management Statistics	
Display SNMP Notification Statistics	
View SNMP notification statistics	<code>show snmp notifies</code>

To do this:	Enter this command:
Display SNMP Access Statistics	
View SNMP access statistics	<code>show snmp accesses</code>
Display SNMP Trap History	
View SNMP trap history	<code>show snmp trap history</code>
Display SNMP Trap Statistics	
View SNMP Trap Statistics	<code>show snmp trap statistics</code>
Display ORBEM Information	
View ORBEM client status	<code>show orbem client id</code>
View ORBEM session information	<code>show orbem session table</code>
View individual ORBEM sessions	<code>show orbem session id orbem</code>
View ORBEM status information	<code>show orbem status</code>
View Port Counters	
Display Port Datalink Counters	
View datalink counters for a specific port	<code>show port datalink counters slot#/port#</code>
Display Port Network Processor Unit (NPU) Counters	
View NPU counters for a specific port	<code>show port npu counters slot#/port#</code>



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.

Monitoring ASR 5000 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.

Table 15. Hardware Monitoring Commands

To do this:	Enter this command:
View the Status of the Power System	
View the status of the PFUs	<code>show power chassis</code>
View the power status of the individual chassis slots	<code>show power all</code>
View the Status of the Fan Trays	
View the status of the fan trays, including current relative speeds and temperatures.	<code>show fans</code>
Determine the Status of Installed Cards	
View a listing of installed application cards	<code>show card table</code>
View a listing of installed line cards	<code>show linecard table</code>
View Line Card-to-Application Card Mappings	
View line card-to-application card mappings	<code>show card mappings</code>
Perform a Hardware Inventory	
View all cards installed in the chassis and their hardware revision, part, serial, assembly, and fabrication numbers	<code>show hardware inventory</code>
View all cards installed in the chassis and their hardware revision, and the firmware version of the on-board Field Programmable Gate Array (FPGAs)	<code>show hardware version board</code>
View details of a specific card. Output contains same information as output of both show hardware inventory and show hardware version board	<code>show hardware card slot_number</code>
View Card Diagnostics	
View boot, power and temperature diagnostics	<code>show maximum-temperature; show card diag slot_number</code>
View runtime, or real time, information	<code>show card info slot_number</code>
View the LED Status of All Installed Cards	
View the LED status for all installed cards	<code>show leds all</code>
View Available Physical Ports	

To do this:	Enter this command:
View ports that are available to the system	<code>show port table</code>
View detailed information for a specific port	<code>show port info</code> <i>slot_number/port_number</i>
View CPU Resource Information	
View CPU resource information	<code>show resource cpu</code>
View CPU resources	<code>show resources { cpu session }</code>
View CPU usage information	<code>show cpu table; show cpu info</code>
View Component Temperature Information	
View current component temperatures	<code>show temperature</code>
View maximum temperatures reached since last timestamp.	<code>show maximum-temperatures</code>

Chapter 9

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 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:

```
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 following is the primary command used to configure the type of schema and the statistics collected:

```
name schema format format_string
```

Refer to the *Bulk Statistics Configuration Mode Commands* chapter and the *Bulk Statistics File Configuration Mode Commands* chapter 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 December 7 12:14:30 EDT 2011

Last successful tx recs: 190

Last successful tx bytes: 13507

Last attempted transfer: Wednesday December 7 12:14:30 EDT 2011

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* subsection of the *Configuring Communication With the Collection Server* section 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: Monday February 14 15:12:30 EST 2011

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: Monday February 14 15:12:30 EST 2011

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 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 outside of the configured sampling interval, enter the following command:

```
bulkstats force gather
```

To manually initiate the transferring of bulk statistics 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 “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 captures 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 16. Logging Events Pertaining to Bulk Statistics

Event	Event ID	Severity	Additional Information
Local File Open Error	31002	Warning	"Unable to open local file <i>filename</i> for storing bulkstats data"
Receiver Open Error	31018	Warning	"Unable to open url <i>filename</i> for storing bulkstats data"
Receiver Write Error	31019	Warning	"Unable to write to url <i>filename</i> while storing bulkstats data"
Receiver Close Error	31020	Warning	"Unable to close url <i>filename</i> while storing bulkstats data"

Chapter 10

Configuring and Viewing System Logs

This chapter describes how to configure parameters related to the various types of logging and how to viewing their content. It includes the following sections:

- [Configuring Event Logging Parameters](#)
- [Configuring Trace Logging](#)
- [Configuring Active Logs](#)
- [Configuring Monitor Logs](#)
- [Viewing Logging Configuration and Statistics](#)
- [Viewing Event Logs Using the CLI](#)
- [Configuring and Viewing Software Crash Logging Parameters](#)
- [Saving Log Files](#)
- [Event ID Overview](#)

System Log Types

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. Configurability depends 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 that 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 operator configurable on a CLI instance-by-CLI instance basis. 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 events 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. Monitors can be performed based on a subscriber's MSID or username.
- **Crash:** Crash logging stores useful information pertaining to system software crashes. They are useful in determining the cause of the crash.

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 entries.

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.
- *Optional:* 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. 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 *Context Configuration Mode Commands* chapter in 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 in the Exec mode. 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 impact session processing. They should be implemented for debug purposes only.

Use the following example to configure trace logs in the Exec mode:

```
logging trace { callid call_id | ipaddr ip_address | msid ms_id | username  
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 | username  
username }
```

Configuring Active Logs

Active logs are event logs that are operator configurable on a CLI instance-by-CLI instance basis. 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 execute the following command in the Global Configuration mode:

```
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 (Global Configuration mode):

```
logging filter active facility facility level report_level
```

```
logging active
```

Notes:

- Refer to the **logging filter** command in the *Global Configuration Mode Commands* chapter of 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 *Exec Mode Commands (D-S)* chapter of 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 in the Exec mode:

```
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 compliance with law enforcement agency requirements for monitoring capabilities of particular subscribers.

Monitors 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
```

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 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 *Exec Mode Show Commands (H-L)* chapter of 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 SMC
- **PCMCIA Flash Card:** Installed 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 written with unique names as they occur to the specified location. The name format is *crash-card-cpu-time-core*. Where *card* is the card slot, *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 in the Global Configuration mode:

```
configure
    crash enable [ encrypted ] url <crash_url>
end
```

Notes:

- Keyword and variable options are available for the **crash enable** command. Refer to the *Global Configuration Mode Commands* chapter of 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 flash memory on the SMC. They are located in the **/flash/crash/** directory with file names in the *mc-slot-cpu-pid-xxxxxxx* 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 *xxxxxxx* 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 2011-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.

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.
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 card where the crash occurred.

Step 2 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 is 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()
[ 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: 2011-Dec-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_verbosity] [facility facility] [level severity_level] [pdu-data pdu_format] [pdu-  
verbosity pdu_verbosity] [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: The use of event IDs depends on the platform type and the licenses running on the platform.

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 17. 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
Active Charging Service Session Manager (ACSMGR) Signalling Interface Logging Facility	97000-97099
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
BSSAP+ Service Facilities	131000- 131199
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

Facility	Event ID Range
Daughter Card Controller Facility Events	62000-62999
Daughter Card Manager Facility Events	57000-57999
DHCP Facility Events	53000-53999
Diameter Accounting Protocol Facility	112000-112999
Diameter Authentication Protocol Facility	111000-111999
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
Femto Network Gateway Facility Events	149000-149999
Firewall Facility	96000-96999
Foreign Agent Manager Facility Events	33000-33999
GCDR Facility Events	66000-66999
GPRS Application Facility (GPRS)	115100-115399
GTP-PRIME Protocol Facility Events	52000-52999
GTPC Protocol Facility Events	47000-47999
GTPC Signaling Demultiplexer 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
IKEv2 Facility Events	122000-122999
IMS Authorization Service Library Facility Events	98100-98999
IMSI Manager Facility	114000-114999
IMS SH Library Events	124000-124999
IMSUE Logging Facility	135000 -136999
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

Facility	Event ID Range
IP Pool Sharing Protocol (IPSP) Facility Events	68000-68999
IP Route Facility Events	20000-20999
IPSG Driver Facility	128000-128999
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
Link Manager Facility	89500-89999
MEGADIAM Session Manager service (MEGADIAM) Facility	121200-121999
MEGADIAM Manager (MEGADIAMMGR) Facility	12100 - 121199
MME App Facility Events	147000-147999
MME Demux Manager Facility Events	154000-154999
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
MSEG Application Facility mseg-gtpc Events	172000-172199
MSEG Application Facility mseg-gtpu Events	172200-172299
MSEG Application Facility mseg-app Events	172300-172999
MSEG Manager (MSEGMGR) Facility Events	171000-171999
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

Facility	Event ID Range
PDN Gateway Facility Events	139000-139999
Point-To-Point Protocol Facility Events	25000-25999
P2P Facility	146000-146999
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
SCTP Protocol Facility	87300-87499
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
SGTPC Manager Facility	117000-117999
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
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

Facility	Event ID Range
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 or 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.

```
2011-Dec-11+5:18:41.993 [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 18. Event Element Descriptions

Element	Description
2011-Dec-11+5:18:41.993	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 11

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.

Refer to the *ASR 5000 Installation Guide* for comprehensive descriptions of the hardware components addressed by these troubleshooting procedures.

The following topics are included:

- [Detecting Faulty Hardware](#)
- [Taking Corrective Action](#)
- [Verifying Network Connectivity](#)
- [Using the System Diagnostic Utilities](#)

Detecting Faulty Hardware

When power is applied to the chassis, power is sequentially applied to management cards, application cards and line cards.

Each PFU, application and line card installed in the system incorporates light emitting diodes (LEDs) that indicate its operating status. This section describes how to use these status LEDs to verify that all of the installed components are functioning properly.



Important: As the system progresses through its boot process, some cards will not exhibit immediate LED activity. Allow several minutes to elapse after a reboot is initiated before checking the LEDs on the various cards to verify that the boot process has successfully completed.

Using the CLI to View Status LEDs

The status of application and line card LEDs can be viewed through the CLI by entering the **show leds all** command in Exec mode.

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
```

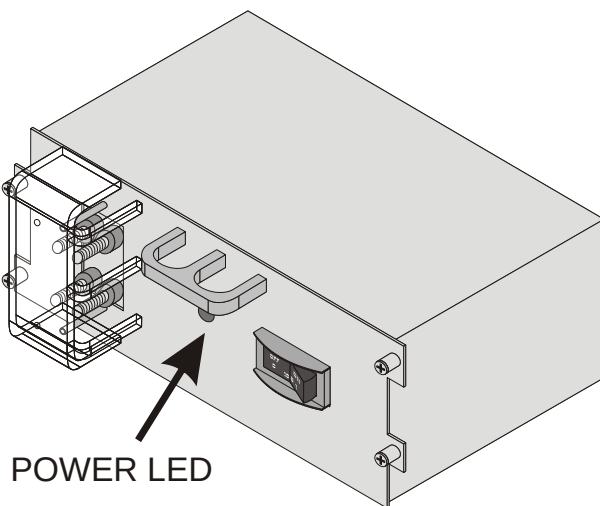
The status of the two Power Filter Units (PFUs) can be viewed by entering the **show power chassis** command in the Exec mode.

Checking the LED on the PFU

Each PFU has a single status LED labeled *POWER*.

This LED should be green for normal operating conditions.

Figure 11. 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 19. PFU POWER LED States

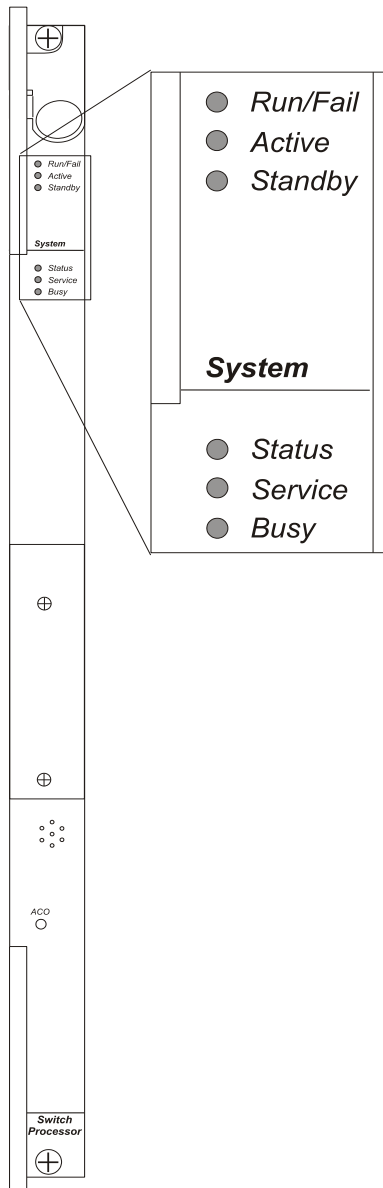
Color	Description	Troubleshooting
Green	PFU powered with no errors detected	None needed.
None	PFU is not receiving power	• Verify that the power switch is in the ON position. • Verify that the RTN and -VDC lugs are attached properly. • Verify that the ground lug is attached properly. Verify that the 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 power distribution panel (PDP) is installed between the power distribution frame (PDF) and the chassis, verify that the circuit breakers are set to ON. • If a PDP is installed between the PDF and the chassis, check the cables from the PDP 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 SMC

Each SMC is equipped with the following LEDs as shown in the accompanying figure:

- Run/Fail
- Active
- Standby
- Status
- Service
- Busy

Figure 12. 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 20. 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 Power On Self Tests (POSTs). It is likely that the errors were logged to the command line interface during boot. Refer to one or more of the following to help analyze this problem: • The <i>Monitoring the System</i> chapter in this guide for show commands, the outputs of which will assist in further determining the problem. • The <i>Configuring and Viewing System Logs</i> chapter in this guide for information on how to view logs. • The <i>SNMP MIB Reference</i> for information on associated status and alarm conditions.
None	Card is not receiving power.	• Verify that the <i>POWER</i> LEDs on the PFUs are green. If they are not, refer to the Checking the LED on the PFU section 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 <i>ASR 5000 Installation Guide</i>. • 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 21. 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.
Blinking Green	Tasks or processes are being migrated from the active SMC to the redundant/secondary SMC.	Verify that the <i>Standby</i> 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 <i>Monitoring the System</i> chapter in this guide for show commands, the outputs of which will assist in further determining the problem. - The <i>Configuring and Viewing System Logs</i> chapter in this guide for information on how to view logs. - The <i>SNMP MIB Reference</i> for information on associated status and alarm conditions.
None	Card is not receiving power. OR Card is in Standby Mode.	- Verify that the <i>Run/Fail</i> 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 for troubleshooting information. - Check the state of the <i>Standby</i> LED. If it is green, the card is in standby mode. If needed, refer to the <i>Configuring Packet Processing Card and Line Card Availability</i> section of the <i>Configuring System Settings</i> chapter in this guide 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 22. 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.
Blinking Green	Tasks or processes are being migrated from the active SMC to the redundant/secondary SMC.	Verify that the <i>Active</i> 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 <i>Monitoring the System</i> chapter in this guide for show commands, the outputs of which will assist in further determining the problem. - The <i>Configuring and Viewing System Logs</i> chapter in this guide for information on how to view logs. - The <i>SNMP MIB Reference</i> for information on associated status and alarm conditions.
None	Card is not receiving power. OR Card is in Active Mode.	- Verify that the <i>Run/Fail</i> 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 for troubleshooting information. - 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 Management Card Switchover section 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 23. SMC Status LED States

Color	Description	Troubleshooting
Green	No system errors detected.	None needed.
Red	Failures detected.	- Check the <i>Run/Fail</i> 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 <i>Monitoring the System</i> chapter in this guide for show commands, the outputs of which will assist in further determining the problem. - The <i>Configuring and Viewing System Logs</i> chapter in this guide for information on how to view logs. - The <i>SNMP MIB Reference</i> for information on associated status and alarm conditions.
None	Card is not receiving power.	Verify that the <i>Run/Fail</i> 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 for troubleshooting information.

SMC Service LED States

The **Service** LEDs on the SMCs indicate that the system requires maintenance or service (for example, the system could not locate 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 24. 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 <i>Monitoring the System</i> chapter in this guide for show commands, the outputs of which will assist in further determining the problem. - The <i>Configuring and Viewing System Logs</i> chapter in this guide for information on how to view logs. - The <i>SNMP MIB Reference</i> 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 25. 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. NOTE: You should wait until this LED is off before removing the SMC from the chassis. This practice ensures the integrity of all data being transferred to or from the memory device.
None	The memory devices are not in use.	No maintenance needed.

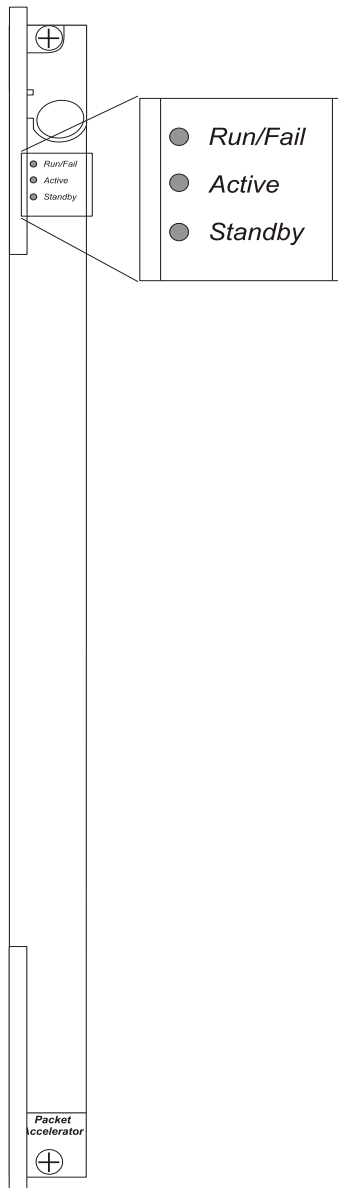
Checking the LEDs on the Packet Processing Cards

The ASR 5000 supports a variety of packet processing cards (PSC n and PPC). For detailed information about the types of cards and their applications, refer to the *ASR 5000 Installation Guide*.

Each packet processing card is equipped with the status LEDs listed below:

- Run/Fail
- Active
- Standby
- Status
- Service

Figure 13. Packet Processing Card LEDs



The possible states for all packet processing card LEDs are described below.

Packet Processing Card Run/Fail LED States

The packet processing card *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 26. Packet Processing Card 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 Power On Self Tests (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 <i>Monitoring the System</i> chapter in this guide for show commands, the outputs of which will assist in further determining the problem. • The <i>Configuring and Viewing System Logs</i> chapter in this guide for information on how to view logs. • The <i>SNMP MIB Reference</i> for information on associated status and alarm conditions.
None	Card is not receiving power.	• Verify that the <i>POWER</i> LEDs on the PFUs are green. If they are not, refer to the Checking the LED on the PFU section 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 <i>ASR 5000 Installation Guide</i>. • If all of the above suggestions have been verified, it is possible that the packet processing card is not functional. Please contact your service representative.

Packet Processing Card Active LED States

The *Active* LED on a packet processing card 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 packet processing cards are booted into standby mode. The system must then be configured as to which packet processing cards should serve as redundant components (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 27. Packet Processing Card Active LED States

Color	Description	Troubleshooting
Green	Card is active.	The first time power is applied to the system, all of the packet processing cards should be booted into the standby mode. Therefore, this LED should be off.
Blinking Green	Tasks or processes are being migrated from an active card to a redundant/secondary card.	Verify that the <i>Standby</i> LED on a redundant packet processing card is also blinking green. If so, there is an issue with the card that was active and is transferring its processes. Refer to the <i>Monitoring the System</i> chapter of this guide for information on determining the status of the packet processing card and system software processes.
None	Card is not receiving power. OR Card is in Standby Mode.	- Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, please refer to the Packet Processing Card Run/Fail LED States section for troubleshooting information. - Check the state of the <i>Standby</i> 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 <i>Configuring Packet Processing and Line Card Availability</i> section of the <i>Configuring System Settings</i> chapter in this reference for information on making the card active.

Packet Processing Card Standby LED States

The *Standby* LED on a packet processing card 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 packet processing cards are booted into standby mode. The system must then be configured as to which packet processing cards should be redundant (remain in standby mode) and which should be active.

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. Packet Processing Card Standby LED States

Color	Description	Troubleshooting
Green	Card is in redundant mode.	The first time power is applied to the system, all of the packet processing cards should be booted into the standby mode. Therefore, this is normal operation.
Blinking Green	Tasks or processes are being migrated from the active SMC to the redundant/secondary SMC.	Verify that the <i>Active</i> LED on the redundant packet processing card is also blinking green. If so, there is an issue with the active packet processing card and the system is transferring its processes. Refer to one or more of the following to help analyze this problem: • The <i>Monitoring the System</i> chapter in this guide for show commands, the outputs of which will assist in further determining the problem. • The <i>Configuring and Viewing System Logs</i> chapter in this guide for information on how to view logs. • The <i>SNMP MIB Reference</i> for information on associated status and alarm conditions.
None	Card is not receiving power. OR Card in Active Mode.	• Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, please refer to the Packet Processing Card Run/Fail LED States section for information on troubleshooting. • Check the state of the <i>Active</i> LED. If it is green, the card is in active mode. If needed, refer to the Manually Initiating a Packet Processing Card Migration section for information on configuring the card to serve as a redundant component.

Checking the LEDs on the SPIO

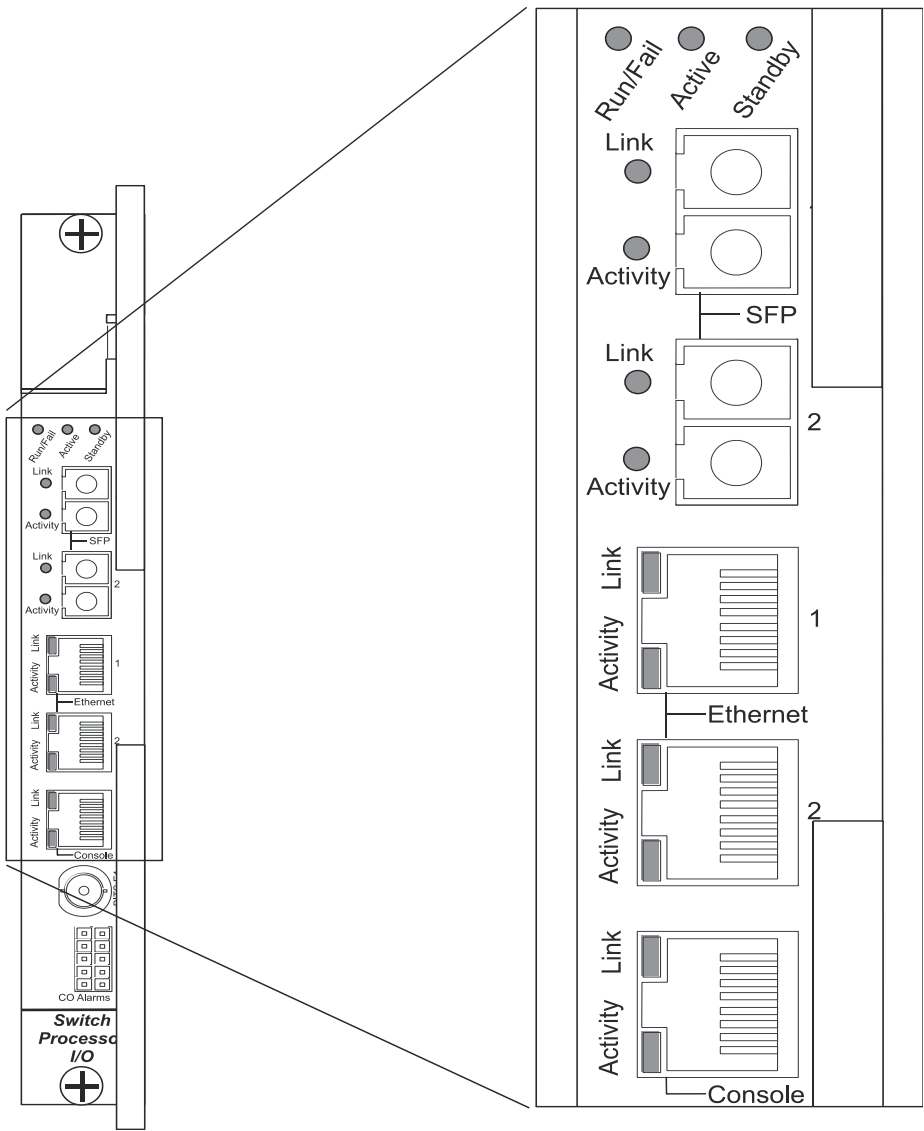
Each SPIO is equipped with the following status LEDs:

- 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 14. SPIO LED Locations



The possible states for all of the SPIO LEDs are described in the sections that follow.

SPIO Run/Fail LED States

The SPIO *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 29. 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 Power On Self Tests (POSTs). It is likely that the errors were logged to the command line interface during the boot process. Refer to the <i>Monitoring the System</i> chapter of this guide for information on determining the status of system hardware components.
None	Card is not receiving power.	• Verify that the <i>POWER</i> LEDs on the PFUs are green. If they are not, refer to the Checking the LED on the PFU section 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 <i>ASR 5000 Installation Guide</i>. • 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 30. 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.
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 for troubleshooting information. - Check the state of the <i>Standby</i> 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 31. 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 SMC installed in slot 8. If the 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 <i>Monitoring the System</i> chapter in this guide for show commands, the outputs of which will assist in further determining the problem. - The <i>Configuring and Viewing System Logs</i> chapter in this guide for information on how to view logs. - The <i>SNMP MIB Reference</i> for information on associated status and alarm conditions.
None	Card is not receiving power. OR Card is in Active Mode.	- Verify that the <i>Run/Fail</i> 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 for troubleshooting information. - Check the state of the <i>Active</i> LED. If it is green, the card is in active mode. This is normal for the SPIO in slot 24 since the chassis automatically makes 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 32. SPIO Interface – Link LED States

Color	Description	Troubleshooting
Green	Link is up.	None needed. NOTE: 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 is available to card. OR Link is down.	- Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power. If it is off, refer to the SPIO Run/Fail LED States section 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 33. 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 Ethernet Line Cards

The ASR 5000 can be equipped with a variety of Ethernet line cards that support subscriber traffic. For detailed information about the types of line cards and their applications, refer to the *ASR 5000 Installation Guide*.

The following line cards are currently supported on the ASR 5000:

- Fast Ethernet Line Card (FELC and FLC2): half-height, 8-ports, 10/100Base-T interfaces
- Gigabit Ethernet Line Card (GELC and GLC2): half-height, 1-ports, SFP interface
- Quad GigE Line Card (QGLC): half-height, 4-ports, SFP interfaces
- 10 Gigabit Line Card (XGLC): full-height, 1-port, SFP+ interfaces

Each of the Ethernet cards listed above 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 *Link* and *Activity* LEDs.

The possible states for all LEDs on these Ethernet line cards are described below.

Ethernet Line Card Run/Fail LED States

The *Run/Fail* LEDs on the Ethernet 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 34. Ethernet Line Card 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 Power On Self Tests (POSTs). It is likely that the errors were logged to the command line interface during the boot process. Refer to the <i>Monitoring the System</i> chapter of this guide for information on determining the status of system hardware components.
None	Card is not receiving power.	• Verify that the <i>POWER</i> LEDs on the PFUs are green. If they are not, refer to the Checking the LED on the PFU section 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 <i>ASR 5000 Installation Guide</i>. • 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 Line Card Active LED States

The *Active* LEDs on the Ethernet line cards indicate that the operating software is loaded on the card and that the card is ready for operation.

 **Important:** QGLCs and XGLCs only work in an ASR 5000 behind specific types of packet processing cards. Refer to the *ASR 5000 Installation Guide* for details.

The line cards will remain in a ready mode until their corresponding packet processing card is made active via configuration. While in ready mode the *Active* LED should be off. After the packet processing card is made active, the line card installed in the upper-rear chassis slot behind the packet processing card will also be made active. The line card (except for the Full-height XGLC) installed in the lower-rear chassis slot behind the packet processing card 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 35. Ethernet Line Card 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 half-height 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.
None	Card is in Ready Mode. OR Card is not receiving power. OR Card is in Standby Mode.	- This is normal prior to configuration. Neither the <i>Active</i> or the <i>Standby</i> LED on the card is on. - Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the Ethernet Line Card Run/Fail LED States section for troubleshooting information. - Check the state of the <i>Standby</i> 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 <i>Configuring Packet Processing and Line Card Availability</i> section of the <i>Configuring System Settings</i> chapter in this guide for information on making the card active.

Ethernet Line Card Standby LED States

The *Standby* LEDs on the Ethernet line cards indicate that software is loaded on the cards, but are serving as redundant components.

The line cards will remain in a ready mode until their corresponding packet processing card is made active via configuration. While in ready mode, the Active LED should be off. After the packet processing card is made active, the line card installed in the upper-rear chassis slot behind the packet processing card will also be made active. The line card (except for the full-height XGLC) installed in the lower-rear chassis slot behind the packet processing card 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 36. Ethernet Line Card Standby LED States

Color	Description	Troubleshooting
Green	Card is in redundant mode.	None needed for half-height 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 <i>Monitoring the System</i> chapter of this guide for information on determining the status of the line card and system software processes.
None	Card is in Ready Mode. OR Card is not receiving power. OR Card is in Active Mode.	- This is normal prior to configuration. Neither the Active nor <i>Standby</i> LEDs on the card is on. - Verify that the <i>Run/Fail</i> LED is green. If so, the card is receiving power and POST test results are positive. If it is off, refer to the Ethernet Line Card Run/Fail LED States section for troubleshooting information. - Check the state of the <i>Active</i> LED. If it is green, the card is in standby mode. If needed, refer to the Manually Initiating a Line Card or SPIO Switchover section for information on configuring the card to serve as a redundant component.

Ethernet Line Card Interface – Link LED States

The *Link* LEDs, associated with a particular network interface on the Ethernet 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 37. Ethernet Line Card Interface – Link LED States

Color	Description	Troubleshooting
Green	Link is up.	None needed. NOTE: 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 is available to the card. OR Link is down.	- Verify that the <i>Run/Fail LED</i> is green. If so, the card is receiving power. If it is off, refer to the Ethernet Line Card 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 Line Card Interface Activity LED States

The *Activity* LEDs, associated with a particular network interface on the Ethernet 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 38. Ethernet Line Card Interface Activity LED States

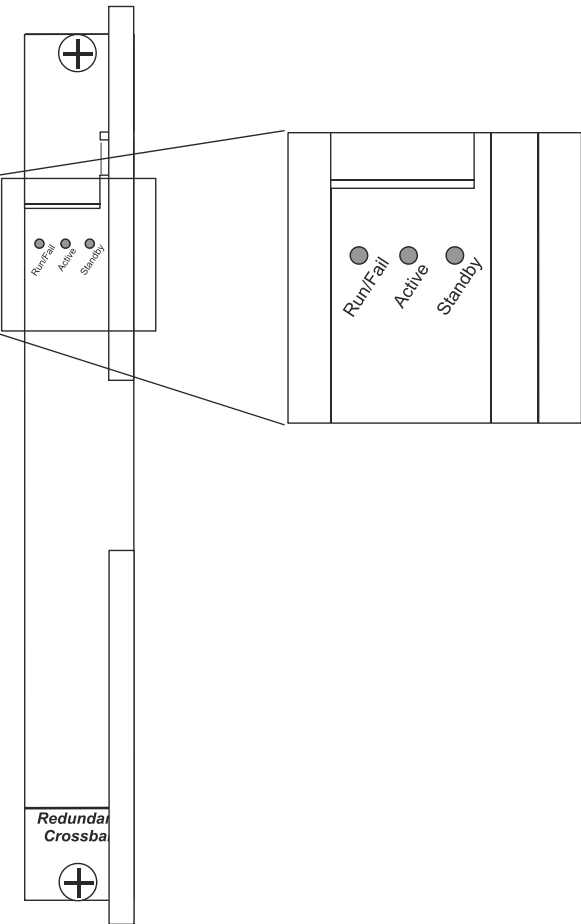
Color	Description	Troubleshooting
Flashing Green	Traffic is present on the link.	None needed.
None	No traffic is present on the link.	None needed if there is no activity on the link. Prior to configuration, this is normal operation.

Checking the LEDs on the RCC

Each RCC is equipped with status LEDs as listed below:

- Run/Fail
- Active
- Standby

Figure 15. RCC LED Locations



The possible states for all of the RCC LEDs are described in the sections that follow.

RCC Run/Fail LED States

The *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 39. RCC 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 Power On Self Tests (POSTs). It is likely that the errors were logged to the command line interface during the boot process. Refer to the <i>Monitoring the System</i> chapter of this guide for information on determining the status of system hardware components.
None	Card powered with error(s) detected.	- Verify that the <i>POWER</i> LEDs on the PFUs are green. If they are not, refer to the Checking the LED on the PFU section 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 <i>ASR 5000 Installation Guide</i>. - If all of the above suggestions have been verified, it is possible that the RCC 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 40. RCC Active LED States

Color	Description	Troubleshooting
Green	Card is active.	The RCC is actively routing traffic from a line card installed behind a packet processing card that has failed to a redundant packet processing card. 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 the Checking the LEDs on the Packet Processing Cards section to determine which card has failed.
None	Card is not receiving power. OR Card is in Standby Mode.	- Verify that the <i>Run/Fail</i> 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 for troubleshooting information. - Check the state of the <i>Standby</i> 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 packet processing card. 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 41. 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 is in Active Mode.	- Verify that the <i>Run/Fail</i> 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 for troubleshooting information. - Check the state of the <i>Active</i> 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 packet processing card that has failed. - Refer to the Checking the LEDs on the Packet Processing Cards section to determine which packet processing card has failed. Information on determining the cause of the failure can be found in the <i>Monitoring the System</i> chapter of this guide.

Testing System Alarm Outputs

The system provides the following two physical alarm mechanisms:

- **System Audible Alarm:** Located on the 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 dry-contact relays (Form C) 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 SMC to verify operation.
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 SMC installed in chassis slot 8 will boot into the “active” mode and begin booting other system components. The SMC installed in chassis slot 9 will automatically be booted into “standby” mode dictating that it will serve as a redundant component. However, the active SMC will frequently synchronize currently running tasks or processes with the redundant SMC.

In the event of a critical failure on the SMC in slot 8, system control will be switched to the redundant SMC in slot 9. This is a relatively seamless transition because the two are synchronized. The formerly active 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 SMC switch over by entering the following command:

```
card smc switchover

card switch from <24 or 25> to <25 or 24>
```

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 SMC just switched. Its state should be *Standby*.

Manually Initiating a Packet Processing Card Migration

When the system boots up, all packet processing cards 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 the critical failure of a packet processing card, tasks will be automatically be migrated from the active card to a redundant card in standby mode. The line card installed behind the packet processing card 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 packet processing card. Therefore, redundant packet processing cards 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 packet processing card migration. These instructions assume you are at the root prompt for the Exec mode:

```
[local]host_name#
```

Step 1 Initiate a manual packet processing card migration by entering the following command:

```
card psc migration from original_slot# to final_slot#
```

```
card migrate from original_slot# to final_slot#
```

Keyword/Variable	Description
from	Specifies the chassis slot number of the packet processing card that is being migrated from <i>original_slot</i> which is an integer from 1 through 7, and 10 through 16.
to	Specifies the chassis slot number of the packet processing card that is being migrated to <i>final_slot</i> which is an integer from 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 packet processing card that was just migrated from. Its state should be *Standby*. The state of the packet processing card migrated to should be *Active*.

Manually Initiating a Line Card or SPIO Switchover

Ethernet line cards are installed in the half-height slots at the rear of the chassis. This design allows for two half-height line cards to be installed behind every application card (vertical redundancy). With two line cards installed, booting their associated application card causes the card in the upper-rear chassis slot to 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.

The XGLC is a full-height card that supports 1:1 side-by-side redundancy. Side-by-side (horizontal) redundancy allows two XGLC cards installed in neighboring slots to act as a redundant pair. Side-by-side pair slots for the XGLC are: 17-18, 19-20, 21-22, 23-26, 27-28, 29-30, and 31-32. If the XGLCs are not configured for side-by-side redundancy, they run independently without redundancy.

When configured for side-by-side redundancy, The XGLC is referenced only by the upper slot number (17 through 23, 26 through 33); the lower slot number is not recognized. All other configuration commands work as if the side-by-side slots were top-bottom slots. Configuration commands directed at the bottom slots either fail with errors or are disallowed.

In the event that a SPIO experiences a failure, the system will automatically switch traffic to the redundant SPIO installed behind the redundant SMC.

In the event that an issue arises that is not severe enough for the system to perform an automatic switchover, a manual switchover can be performed. Follow the instructions below to manually initiate a line card or SPIO switchover. 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 switch from slot# to slot#
```

Keyword/Variable	Description
from	Specifies that a switchover will be done for the card in the specified slot number.
to	Specifies the chassis slot number to which interfaces will be migrated. Executing this command will switch network connections from the active card that corresponds to the card being migrated to.
<i>slot#</i>	This can be any of the following integer values: - Half-height – 17 through 23, 26 through 39, or 42 through 48 - Full-height (XGLC) – 17 through 23, 26 through 32

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

Packet processing cards or line cards that are in either the active or standby modes can be halted. Halting these cards places them into the “offline” mode. In this mode, the card is 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 to a redundant component prior to entering the offline mode.

This section describes how to initiate a card halt and restore 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 card 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 initiate the halt operation.

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 **show card table** command at the prompt.

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. Contexts act like virtual private networks (VPNs) that operate independently of other contexts. Ports, interfaces, and routes configured in one context cannot be tested from another context without additional configuration.

To switch between contexts enter the following command at the root prompt for the Exec mode:

```
context context_name
```

context_name is the name of the context to which you wish to switch. The following prompt appears:

```
[context_name]host_name#
```

Using the ping Command

The ICMP **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
<i>host_ip_address</i>	Identifies the remote node to which is the target of the ping command. <i>host_ip_address</i> specifies the remote node using its IP address entered in IPv4 dotted-decimal format.
count <i>num_packets</i>	Specifies the number of packets to send to the remote host for verification. <i>num_packets</i> must be within the range 1 through 10000. The default is 5.
pattern <i>packet_pattern</i>	Specifies a pattern to use to fill the internet control message protocol packets with. <i>packet_pattern</i> must be specified in hexadecimal format with a value in the range hexadecimal 0x0000 through 0xFFFF. <i>packet_pattern</i> must begin with a '0x' followed by up to four hexadecimal digits. The default is that each octet of the packet is encoded with the octet number of the packet.
size <i>octet_count</i>	Specifies the number of bytes each IP datagram. <i>octet_count</i> must be a value in the range 40 through 18432. The default is 56.
src { <i>src_host_name</i> <i>src_host_ip_address</i> }	Specifies an IP address to use in the packets as the source node. <i>src_host_name</i> specifies the source node using the node's logical host name which must be resolved via DNS lookup. <i>src_host_ip_address</i> : specifies the source node using its IP address entered in IPv4 dotted-decimal format. 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 the configuration of the ports and interfaces within the context are correct.
- If the 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 to further troubleshoot the issue (see below).

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
<i>host_name</i>	Identifies the remote node to trace the route to by the hostname. <i>host_name</i> specifies the remote node using the node's logical host name which must be resolved via DNS lookup.
<i>host_ip_address</i>	Identifies the remote node to trace the route to by its IP address. <i>host_ip_address</i> is expressed in IPv4 dotted-decimal notation.
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 <i>max_ttl</i>	Specifies the maximum time to live (TTL) for the route tracing packets. The time to live is the number of hops through the network; it is not a measure of time. <i>max_ttl</i> is an integer from 1 through 255. It is an error if <i>max_ttl</i> is less than <i>min_ttl</i> , whether <i>min_ttl</i> is specified or defaulted. The default maximum TTL is 30 hops.
minttl <i>min_ttl</i>	Specifies the minimum time to live for the route tracing packets. The time to live is the number of hops through the network; it is not a measure of time. <i>min_ttl</i> is an integer from 1 through 255. It is an error if <i>min_ttl</i> is greater than <i>max_ttl</i> , whether <i>max_ttl</i> is specified or defaulted. The default minimum TTL is 1 hop.
port <i>port_number</i>	Specifies a specific port to connect as an integer from 1 through 65535. The default port is 33434.
size	Specifies the number of bytes each packet as an integer from 40 through 32768. The default is 40.
src { <i>src_host_name</i> <i>src_host_ip_address</i> }	Specifies an IP address to use in the packets as the source node. <i>src_host_name</i> specifies the remote node using the node's logical host name which must be resolved via DNS lookup. <i>src_host_ip_address</i> specifies the remote node using its IP address in IPv4 dotted-decimal format. The default is the IP address of the interface through which the ping was issued.
timeout <i>seconds</i>	Specifies the maximum time to wait (in seconds) for a response from each route tracing packet as an integer from 2 through 100. The default is 5.

The following displays a sample output.

```
tracert 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
<i>route_ip_address</i>	Specifies the IP address of a network node for which route information is displayed. The address can be entered in IPv4 dotted-decimal or IPv6 colon-separated format.
<i>route_gw_address</i>	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. The address can be entered in IPv4 dotted-decimal or IPv6 colon-separated format.

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. The address can be entered in IPv4 dotted-decimal or IPv6 colon-separated format. If this keyword is not specified, all entries within the context's ARP table are displayed.



Important: Restarting the VPN Manager removes all interfaces from the kernel which in turn removes all ARP entries. However, the NPU still retains all of the ARP entries so that there is no traffic disruption. 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 SPI01

10.0.4.7 ether 00:05:47:02:03:36 C SPI01

10.0.4.1 ether 00:01:30:F2:7F:00 C SPI01

Using the System Diagnostic Utilities

The system provides protocol monitor and test utilities that are useful when troubleshooting or verifying configurations. The information generated by these utilities can help identify the root cause of a software or network configuration issue.

This section describes how to use these utilities.

Using the Monitor Utility

For troubleshooting purposes, the system provides a protocol monitoring utility. This tool displays protocol information for a particular subscriber session or for every session being processed.

 **Caution:** The monitor tool may cause session processing delays and/or data loss. Therefore, it should be used only when troubleshooting.

Using the Protocol Monitor

The protocol monitor displays information for every 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. You should enable logging on your terminal client 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 from the Exec mode by entering the **monitor protocol** command.
An output listing all the currently available protocols, each with an assigned number, is displayed.
- Step 2** Choose the protocol that you wish to monitor by entering the associated number at the *Select:* prompt. A right arrow (>) 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.

```
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)
Y - Multi-Call Trace ((OFF))

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 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 from the Exec mode enter the **monitor subscriber** command.

```

monitor subscriber { callid | imei | imsi | ipaddr | msid | | msisdn | next-call
| pcf | peer-fa | peer-lac | sgsn-address | type | username }

```

- Step 2** Specify the method the monitor should use by entering the appropriate keyword.

- Step 3** Select other options and/or enter the appropriate information for the selected keyword.

If no session matching the specified criteria was being processed when the monitor was invoked, a screen of available monitoring options appears.

- 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 **x** 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 **Enter** 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 help determine the accuracy of the system's DHCP configuration and the server's response time.

This tool provides a mechanism for obtaining an IP address for one or more DHCP servers with which the system communicates.



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 within the appropriate context:

```
dhcp test dhcp-service { service_name } [ all | server | ip_addr ]
```

Table 42. Sample dhcp test dhcp-service Command Output

Keyword	Description
<i>service_name</i>	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 the server under test.

Chapter 12

System Recovery

This chapter describes how to recover a system after it has failed to complete a reboot following a power off cycle or interruption of the normal boot sequence following a **reload** command.

This chapter includes the following sections:

- [Prerequisites](#)
- [Accessing the boot CLI](#)
- [Booting from a Selected Image](#)

 **Caution:** This system recovery process interrupts subscriber service by dropping any existing flows and preventing traffic from being processed during the boot interval. It should only be initiated as an emergency measure.

Prerequisites

Successful recovery from a failed reboot requires that you have access to the system via a console port, and have an uncorrupted copy of the StarOS boot image file stored in flash memory on the SMC, or accessible from an external PCMCIA memory device.

Console Access

The boot recovery sequence can only be executed via a terminal connected to the serial console port on the active SPIO card. This connection can be through a terminal server that is accessible via a LAN interface.

The boot recovery sequence can only be viewed via the console port.

Boot Image

The boot recovery command line interface allows you to specify from which boot image you would like to boot the system. If the system failed to reload following a software update, you can initiate a boot from a previously stored image.

The system recovery procedure will prompt you to enter the path name for the location of the StarOS boot image from which the system will boot. By default the boot command will timeout and attempt to reload the highest priority image from flash memory using the default configuration file.

A boot image binary file name uses the following format:

```
production.build_number.asr5000.bin
```

Refer to the *Configuring the Boot Stack* section in the *Software Management Operations* chapter for additional information on boot stack entries and prioritization.

Accessing the boot CLI

To access the boot CLI you must interrupt an in-progress reload (reboot) sequence.

 **Caution:** This system recovery process interrupts subscriber service by dropping any existing flows and preventing traffic from being processed during the boot interval. It should only be initiated as an emergency measure.

Initiate a Reboot

A reload can be initiated in one of two ways:

- Power cycle the chassis – Turn the circuit breakers on the power filter units (PFUs) Off (O) and then On (I).
- Execute a **reload** command

```
[local]asr5000# reload -noconfirm
```

The boot sequence displays messages on the terminal as it steps through its processes.

Interrupt the Boot Sequence

When the “Booting priority” message line appears (and not before), press CTRL+C to break out of the boot process as shown in the example below:

```
Booting priority 8

image : /flash/production.35272.st40.bin

config: /flash/system.cfg

Entry at 0x00000000cba45e0
```

Press CTRL+C at this point in the sequence.

A message similar to the following appears after the boot process has been interrupted:

```
*****9/0 Ctrl-C Pressed-----

Failed.

aborted by user

8/0:boot>
```

Enter CLI Mode

With the boot prompt displayed, enter **cli** to access the boot recovery CLI. The CLI prompt changes as shown below:

```
8/0:boot>cli
```

```
8/0:cli>
```

boot Command Syntax

The boot recovery command has the following syntax:

```
boot [ -show | -priority=* | -config=* | -noconfig ] { bootfile_URL }
```

The options for this command include:

- **-show**: displays the current boot configuration
- **-priority=***: selects the desired boot stack priority (*)
- **-config=***: enters the desired configuration filename (*), if not the default file
- **-noconfig**: boots using no configuration file

bootfile_URL is the URL for the location of the StarOS boot image file. It specifies the path and file name of the StarOS .bin file from which the system will be booted.

The URL may refer to a local file (flash) or an external file on a PCMCIA device attached to the SMC. The URL must be entered in the following format:

```
{ /flash | /pcmcia1 /filename
```


Booting from a Selected Image

You will issue a boot command via the boot CLI to initiate the system recovery process.

Boot Using No Configuration File

This procedure boots the system using the specified boot image without also loading a configuration file. A sample command string appears below:

```
8/0:cli>boot -noconfig /flash/production.41731.asr5000.bin
```

The boot sequence ends with a prompt to enter the Quick Setup Wizard for creating a configuration file.

```
Launching StarOS

Starting program at 0x0000000000100000

Starent Networks ASR5000 Intelligent Mobile Gateway

SMC is starting up.....

Starting software <version> <build_number>...

No configuration found, press enter to continue.

1. Do you wish to continue with the Quick Setup Wizard[yes/no]:
```

You can exit the Quick Setup Wizard by entering **no** in response to the above prompt. Load a desired configuration file using the Exec mode **configure** command followed by the URL for the configuration file as shown in the example below:

```
[local]asr5000# configure /flash/system.cfg
```

Boot Using A Specified Configuration File

This procedure boots the system using the specified boot image and configuration file. A sample command string appears below:

```
8/0:cli>boot -config=/flash/system.cfg /flash/production.41731.asr5000.bin
```

The boot sequence ends with the appearance of the CLI prompt.

```
[local]asr5000#
```

Confirm that the desired configuration has loaded by running the Exec mode **show configuration** command.

Appendix A

Engineering Rules

This appendix provides engineering guidelines for configuring the system to meet network deployment requirements.

This appendix consists of the following topics:

- [CLI Session Rules](#)
- [ASR 5000 Interface and Port Rules](#)
- [ASR 5000 Packet Processing Card Rules](#)
- [Context Rules](#)
- [Subscriber Rules](#)
- [Service Rules](#)
- [Access Control List \(ACL\) Engineering Rules](#)

CLI Session Rules

Multiple CLI session support is based on the amount of available memory. The Resource Manager reserves enough resources to support a minimum of six CLI sessions 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 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.

ASR 5000 Interface and Port Rules

The rules discussed in this section pertain to the following Ethernet line cards and their interfaces regardless of the application.

- Fast Ethernet 10/100 (FELC/FLC2)
- Gigabit Ethernet 1000 (GELC/GLC2)
- Quad Gigabit Ethernet (QGLC)
- 10 Gigabit Ethernet Line Card (XGLC)

Line Card Rules

The following engineering rules apply to the Fast Ethernet 10/100, Gigabit Ethernet 1000, Quad Gigabit Ethernet and 10 Gigabit Ethernet 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 access control list (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 1,024 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 + 1,017 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 (including the IP address) is transferred to the standby line card.
 - Half-height line cards must installed in the upper and lower chassis slots behind a sa packet processor card must be of the same type: FELC/FLC2, GELC/GLC2, or QGLC 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.

ASR 5000 Packet Processing Card Rules

The following engineering rules apply to the packet processing application cards:

- Configuring a line card to enter the active mode results in the following:
 - The total number of packet processing cards that will become operationally active is increased by one.
 - In the event of a failure, the line card(s) directly behind the packet processing cards will become available directly, or to another packet processing cards via the RCC.
- If you want processing-only application cards, all line card slots directly behind the such cards can be empty. Otherwise, disable those line card slots with the **shutdown** command described in the *Command Line Interface Reference*.
- If you want standby (redundant) packet processing cards, do not populate 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 *Card Configuration Mode Commands* section of the *Command Line Interface Reference*.
- A line card will not handle subscriber traffic unless the packet processing 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 packet processing cards with a total system limit of 5,000 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 64 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.

- 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 1,200.
- 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 (ICSR) 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 1,600 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 (one primary and one secondary per server group) can be configured per context.
- Up to 12 charging gateway functions (CGFs) for GTP 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, you should not configure a large number of services unless your application absolutely requires it. Please contact your Cisco 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 1,024.
- The maximum number of ACL rules per IPSec policy is 1.
 - The maximum number of IPSec ACL rules per context is 1,024.
- 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 ($128 \times 8 \text{ ACLs} = 1,024 \text{ ACL rules per context}$).
- 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 = 1,024 (the ACL rules limit per context)

Appendix B

System Software Task and Subsystem Descriptions

This appendix describes the system and subsystem tasks running under StarOS on an ASR 5000 platform.

It includes the following sections:

- [Overview](#)
- [Primary Task Subsystems](#)
- [Primary Subsystem Controllers and Managers](#)

Overview

For redundancy, scalability and robust call processing, StarOS 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. This distributed design provides fault containment that greatly minimizes the impact to 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 System Management Cards (SMCs) each have a single CPU that is responsible for running tasks related to system management and control. The packet processing cards (PSC n , PPC) contain two CPUs (CPU 0 and CPU 1). These CPUs are responsible for session processing and running the various tasks and processes required to handle mobile data calls. In addition to the CPUs, the packet processing cards 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](#)
- [Primary Subsystem Controllers and Managers](#)

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):** With 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 SMC and synchronizes the information it contains with the RCT subsystem on the standby 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 SMC and synchronizes the information it contains with the SCT subsystem on the standby 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
 - IP forwarding decisions (both unicast and multicast)
 - Per-interface packet filtering
 - Traffic management and traffic engineering
 - Passing user data frames to/from packet processing car 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 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 packet processing cards (through a Redundancy Crossbar Card [RCC], if necessary).

The CSP subsystem runs only on the active 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 Controllers and Managers

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 section provides information about the composition of the primary subsystems that are composed of critical, controller, and / or manager tasks.

ASR 5x00 Subsystems

The following tables describe managers and tasks performing within the specified subsystems on an ASR 5x00 platform.



Important: Variations regarding how the managers and tasks are distributed based on session recovery (SR) are included in the Card and CPU columns in some tables. Tables without these indicators are applicable to ASR 5x00s with and without session recovery. The ASR 5x00 dynamically distributes processes, tasks, and managers on startup. The following tables list the typical locations but variations can occur depending on available resources.

Table 43. ASR 5x00 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 SMCs in either active or standby mode. - Registers tasks with HAT task. - Notifies CSP task of CPU startup completion.	All	All
SITREAP sub-function	Shuts down tasks as required.	All	All

Table 44. ASR 5x00 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 SMC mirrors the HAT subsystem on the active 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. - Initializes and monitors the dedicated hardware on packet processing cards.	All packet processing cards	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 packet processing cards	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 45. ASR 5x00 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 46. ASR 5x00 Virtual Private Networking (VPN) Subsystem

Task	Description	Card		CPU	
		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 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 SMC (local context)	0 (all contexts)	0 (local context)
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. - Maintains the BGP peering connections. - Applies any defined BGP routing policy.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
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. • Maintains the OSPF neighboring relationship. • Maintains the LSA database. • Performs SPF calculations. • Applies 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. • Maintains the OSPFv3 neighboring relationship. • Maintains the LSA database. • Performs OSPFv3 SPF calculations. • Applies any defined OSPFv3 routing policy.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.
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. • Maintains the RIP database. • Sends periodic RIP update messages. • Applies 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. It is responsible for maintaining the routing table for the context. • Maintains the routing table (RIB and FIB). • Performs static routing. • Interfaces to the kernel for routing & interface updates. • Redistributes routing information to dynamic routing protocols. • Calculates nexthop reachability.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.	Same as VPN Mgr.

Table 47. ASR 5x00 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
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 and packet processing cards	0

Table 48. ASR 5x00 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, 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 packet processing cards	All packet processing cards except the first	0	0 on all packet processing cards except the first
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). NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.	Active packet processing cards	First packet processing card	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. - 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. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card. The HNBMgrs should not be started on a packet processing card which has the HNB DEMUX MGR started.	Active packet processing card	Active packet processing card (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 ASNGW Manager can service sessions from multiple ASN PC Managers and multiple contexts. - Protocol processing for R3, R4, R6, GRE tunneling, and Mobile IP	Active packet processing cards		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. - Does protocol processing for R3, R4, R6, GRE tunneling, and Mobile IP	Active packet processing cards		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 packet processing cards	All packet processing cards (except first)	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)	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.	First packet processing card	First packet processing card	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 packet processing cards		0	
Diameter Proxy (DIAMPROXY)	Diameter proxy is created by DIACTRL (which runs as part of VPNCTRL) 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 packet processing cards. When multiple proxies are configured, one DIAMPROXY task is run per packet processing card. 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 (does not maintain any queues). A single Diameter proxy is used to service multiple Diameter applications.	Active packet processing cards (see description)	Active packet processing cards (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. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.	Packet processing card	First packet processing card	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. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.	Packet processing card	First packet processing card	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. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.	Active packet processing cards	First packet processing card	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 packet processing cards		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. • Supports GTPU Echo handling. • Provides Path Failure detection on no response for GTPU echo. • Receives Error-Ind and demuxes it to a particular Session Manager. • Serves as the Default GTPU listener. GTPUMGR will process GTPU packets with invalid TEID. The above features are supported for both GTPUv0 and GTPUv1. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.	Active packet processing cards	First packet processing card	0	Any (see NOTE)
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. • Remains aware of all the active HNB-GW services in the system. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card, but should not be created on the same packet processing card that has HNB Manager.	Packet processing card	Packet processing card (see NOTE)	0	Any (see NOTE)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
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. - Runs the SCTP protocol stack. - Handles the SCTP associations. - Maintains Home-NodeB databases. - Provides nodal functions for Iuh interface on SCTP protocol. NOTE: With session recovery (SR) enabled, this manager is usually established on one of the CPUs on the first active packet processing card. The HNBMgrs should not be started on a packet processing card which has the HNB DEMUX MGR started.	Active packet processing card	Active packet processing card (see NOTE)	0	Any (see NOTE)
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. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.	Active packet processing cards	First packet processing card	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 task. IMSI Manager performs the following functions: • Signaling De-multiplexer: Selects which SessMgr to use for new subscriber sessions. • IMSI-to-SessMgr resolution: Maintains and reports MME-related demux statistics on events like Attach by IMSI, Attach by GUTI, etc. IMSIMgr can interact with the following tasks in the system: • Session Controller • MME Manager • Session Manager NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card. The IMSIMgr will not start on a packet processing card in which SessMgrs are started.	Active packet processing card	Active packet processing card (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. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active deumux packet processing card. The IMSIMgr will not start on a packet processing card in which SessMgrs are already started.	Active packet processing card	Active packet processing card (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 packet processing cards	—	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. - Maintains protocol state machines for all L2TP sessions and tunnels. - Triggers IPSec encryption for new L2TP tunnels as needed. - Works with Session Managers to gracefully bring down tunnels. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.	Active packet processing cards	First packet processing card	0	Any (see NOTE)
L2TP Demultiplexor Task (L2TPDEMUX)	Created by the Session Controller when an 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. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.	Active packet processing cards	First packet processing card.	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. • Supports debugging facility – “magmgr” and “mobile-ipv6”	Packet processing card		Same as VPN Mgr.	
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. • Remains aware of all the active MME services in the system. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the First active packet processing card but should not be created on the same packet processing card that has MME Manager.	Packet processing card	Packet processing card (see NOTE)	0	Any (see NOTE)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
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. - Runs the SCTP protocol stack. - Handles the SCTP associations. - Maintains TA List. - Manage eNodeB databases. - Provides nodal functions for S1-MME protocol. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card. The MMEMGRs should not be started on a packet processing card which has the MME DEMUX MGR started.	Active packet processing card	Active Active packet processing card (see NOTE)	0	Any (see NOTE)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
PCC Bindmux Manager (PCC Demux)	The Bindmux Manager is started as part of PCC service creation procedure. There will be only one BindMux MGR in the chassis. The BindMux function handles multiplexing of the sessions across the available PCCMgrs along with the session binding functions. The capabilities includes: • Monitors load on PCCMgrs. • Distributes incoming IP-CAN connections across PCC Mgrs in the system. • Performs session binding; binds IP-CAN/Gateway session with the AF-Session. • Ensures all messaging for an IMSI across various interfaces is directed towards the selected PCCMgr. • Remains aware of all the active PCC services in the system. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card but should not be created on the same packet processing card that has PCC Manager.	Active packet processing card	Active packet processing card (see NOTE)	0	Any (see NOTE)

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
PCC Manager (PCCMgr)	The PCC Manager is part of the SessMgr process. All PCC Managers will have all the Active PCC Policy Services configured and will be identical in configuration and capabilities. The PCC Manager function manages all the sessions and forms the main block of IPCF functionality which is making authorization decisions. This is the rules engine that acts on the user defined policy profiles; implements rules to generate the policy and charging decision applicable to a subscriber's usage towards an application service. Along with the PCC processing logic, the PCC-Mgr co-hosts the Gx-API and Rx-API modules. It includes the following capabilities: • Runs the Gx/Gxa protocol stack. • Handles the IP-CAN associations. • Provides nodal functions for Gx/Gxa/Rx interface with PCEF/BBERF/AF. NOTE: With session recovery (SR) enabled, this manager is usually established on one of the CPUs on the first active packet processing card. The PCCMGRs should not be started on a packet processing card which has the PCC BINDMUX MGR started.	Active packet processing card	Active packet processing card (see NOTE)	0	Any (see NOTE)
SPR Manager (SPRMgr)	The SPR Manager is part of AAAMgr process and manages the transactions towards an external SSC/SPR node over Sp interface. The Sp interface is based on Sh protocol interface. The SPRMgr abstracts the Sp interactions required for the PCC functions. For more information, refer AAAMgr portion of this table.	Active packet processing card	Active packet processing card (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. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active demux packet processing card. The IMSIMgr will not start on a packet processing card in which SessMgrs are already started.	Active packet processing card	Active Demux packet processing card (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: • Runs as a single instance. • 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. NOTE: With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active demux packet processing card. The IMSIMgr will not start on a packet processing card in which SessMgrs are already started.	Active packet processing card	Active Demux packet processing card (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 packet processing card not running an MMgr	Any Active non- demux packet processing card		

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 packet processing cards 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 and categorizes the URL based 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 packet processing cards			

Table 49. ASR 5x00 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 packet processing cards	All packet processing cards	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 drive 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.	Packet processing card	Packet processing card	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)	Monitors the switch fabric and the gigabit Ethernet control plane.	Packet processing cards	Packet processing cards	0	0

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
Utilities Configuration Manager (UCM)	DHCPD, DNS, FTPD, INETD, NTPD, PING, RLOGIN, SFTPD, SFTP-SERVER, SNMPD, SSH, SSHD, TELNET, TELNETD, TFTP, TRACEROUTE	Active SMC	Active SMC	0	0

Table 50. ASR 5x00 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: - Notifies the Application Servers (EMS) of event occurrences. - Registers such Application Servers (EMS) and subscribes them to associated event types. - 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

Task	Description	Card		CPU	
		w/o SR	w/ SR	w/o SR	w/ SR
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

Appendix C

Access Control Lists

This appendix describes system support for access control lists and explains how they are configured. The product administration guides provide examples and procedures for configuration of basic services on the system. You should select the configuration example that best meets your service model before using the procedures in this appendix.

 **Important:** You do not require a license to configure ACLs. However, the number of ACLs configured may impact performance significantly.

This appendix contains the following sections:

- [Understanding ACLs](#)
- [Configuring ACLs on the System](#)
- [Applying IP ACLs](#)

 **Important:** Not all commands and keywords/variables may be available. Availability depends on the platform type.

Overview

IP access lists, commonly known as access control lists (ACLs), control the flow of packets into and out of the system. They are configured on a per-context basis and consist of “rules” (ACL rules) or filters that control the action taken on packets that match the filter criteria. Once configured, an ACL can be applied to any of the following:

- An individual interface
- All traffic facilitated by a context (known as a policy ACL)
- An individual subscriber
- All subscriber sessions facilitated by a specific context

Understanding ACLs

This section discusses the two main aspects to ACLs on the system:

- Rule(s)
- Rule Order

 **Important:** Refer to the *ACL Configuration Mode Commands* chapter of the *Command Line Interface Reference* for the full command syntax.

Rule(s)

A single ACL consists of one or more ACL rules. Each rule is a filter configured to take a specific action when packets matching specific criteria. Up to 128-rules can be configured per ACL.

 **Important:** Configured ACLs consisting of no rules imply a “deny any” rule. The **deny** action and **any** criteria are discussed later in this section. This is the default behavior for an empty ACL.

Each rule specifies the action to take when a packet matches the specifies criteria. This section discusses the rule actions and criteria supported by the system.

Actions

ACLs specify that one of the following actions can be taken on a packet that matches the specified criteria:

- **Permit:** The packet is accepted and processed.
- **Deny:** The packet is rejected.
- **Redirect:** The packet is forwarded to the specified next-hop address through a specific system interface or to the specified context for processing.

 **Important:** Redirect rules are ignored for ACLs applied to specific subscribers or all subscribers facilitated by a specific context, or APN for UMTS subscribers.

Criteria

Each ACL consists of one or more rules specifying the criteria that packets will be compared against.

The following criteria are supported:

- **Any:** Filters all packets
- **Host:** Filters packets based on the source host IP address
- **ICMP:** Filters Internet Control Message Protocol (ICMP) packets
- **IP:** Filters Internet Protocol (IP) packets
- **Source IP Address:** Filter packets based on one or more source IP addresses

- **TCP:** Filters Transport Control Protocol (TCP) packets
- **UDP:** Filters User Datagram Protocol (UDP) packets

Each of the above criteria are described in detail in the sections that follow.



Important: The following sections contain basic ACL rule syntax information. Refer to the *ACL Configuration Mode Commands* chapter of the *Command Line Interface Reference* for the full command syntax.

- **Any:** The rule applies to all packets.
- **Host:** The rule applies to a specific host as determined by its IP address.
- **ICMP:** The rule applies to specific Internet Control Message Protocol (ICMP) packets, Types, or Codes. ICMP type and code definitions can be found at www.iana.org (RFC 3232).
- **IP:** The rule applies to specific Internet Protocol (IP) packets or fragments.
- **IP Packet Size Identification Algorithm:** The rule applies to specific Internet Protocol (IP) packets identification for fragmentation during forwarding.

This configuration is related to the “IP Identification field” assignment algorithm used by the system, when subscriber packets are being encapsulated (such as Mobile IP and other tunneling encapsulation). Within the system, subscriber packet encapsulation is done in a distributed way and a 16-bit IP identification space is divided and distributed to each entity which does the encapsulation, so that unique IP identification value can be assigned for IP headers during encapsulation.

Since this distributed IP Identification space is small, a non-zero unique identification will be assigned only for those packets which may potentially be fragmented during forwarding (since the IP identification field is only used for reassembly of the fragmented packet). The total size of the IP packet is used to determine the possibility of that packet getting fragmented.

- **Source IP Address:** The rule applies to specific packets originating from a specific source address or a group of source addresses.
- **TCP:** The rule applies to any Transport Control Protocol (TCP) traffic and could be filtered on any combination of source/destination IP addresses, a specific port number, or a group of port numbers. TCP port numbers definitions can be found at www.iana.org
- **UDP:** The rule applies to any User Datagram Protocol (UDP) traffic and could be filtered on any combination of source/destination IP addresses, a specific port number, or a group of port numbers. UDP port numbers definitions can be found at www.iana.org.

Rule Order

A single ACL can consist of multiple rules. Each packet is compared against each of the ACL rules, in the order in which they were entered, until a match is found. Once a match is identified, all subsequent rules are ignored.

Additional rules can be added to an existing ACL and properly ordered using either of the following options:

- Before
- After

Using these placement options requires the specification of an existing rule in the ACL and the configuration of the new rule as demonstrated by the following flow:

```
[ before | after ] { <existing_rule> }
```

Configuring ACLs on the System

This section describes how to configure ACLs.



Important: This section provides the minimum instruction set for configuring access control list on the system. For more information on commands that configure additional parameters and options, refer to the *ACL Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

To configure the system to provide an access control list facility to subscribers:

- Step 1** Create the access control list by applying the example configuration in the [Creating ACLs](#) section.
- Step 2** Specify the rules and criteria for action in ACL list by applying the example configuration in the [Configuring Action and Criteria for Subscriber Traffic](#) section.
- Step 3** *Optional.* The system provides an “undefined” ACL that acts as a default filter for all packets into the context. The default action is to “permit all”. Modify the default configuration for “unidentified” ACLs for by applying the example configuration in the [Configuring an Undefined ACL](#) section.
- Step 4** Verify your ACL configuration by following the steps in the [Verifying the ACL Configuration](#) section.
- Step 5** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command save configuration. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Creating ACLs

To create an ACL, enter the following command sequence from the Exec mode of the system CLI:

```
configure

context <acl_ctxt_name> [ -noconfirm ]

ip access-list <acl_list_name>

end
```

Notes:

- The maximum number of ACLs that can be configured per context is limited by the amount of available memory in the VPN Manager software task. Typically, the maximum is less than 200.

Configuring Action and Criteria for Subscriber Traffic

To create rules to deny/permit the subscriber traffic and apply the rules after or before action, enter the following command sequence from the Exec mode of the system CLI:

configure

```
context <acl_ctxt_name> [ -noconfirm ]

  ip access-list <acl_list_name>

    deny { <ip_address> | any | host | icmp | ip | log | tcp | udp }

    permit { <ip_address> | any | host | icmp | ip | log | tcp | udp }

    after { deny | permit | readdress | redirect }

    before { deny | permit | readdress | redirect }

  end
```

Notes:



Caution: The system does not apply a “deny any” rule, unless it is specified in the ACL. This behavior can be changed by adding a “deny any” rule at the end of the ACL.

- The maximum number of rules that can be configured per ACL varies depending on how the ACL is to be used. For more information, refer to the *Engineering Rules* appendix in this guide.
- Use the information provided in the [Actions](#) and [Criteria](#) sections of this appendix to configure the rules that comprise the ACL. For more information, refer to the *ACL Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

Configuring an “Undefined” ACL

As discussed previously in this chapter the system uses an “undefined” ACL mechanism for filtering the packet(s) in the event that an ACL that has been applied is not present. This scenario is likely the result of a mis-configuration such as the ACL name being mis-typed during the configuration process.

For these scenarios, the system provides an “undefined” ACL that acts as a default filter for all packets into the context. The default action is to “permit all”.

To modify the default behavior for unidentified ACLs, use the following configuration:

configure

```
context <acl_ctxt_name> [ -noconfirm ]

  access-list undefined { deny-all | permit-all }

end
```


Notes:

- Context name is the name of the context containing the “undefined” ACL to be modified. For more information, refer to the *Context Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

Verifying the ACL Configuration

To verify the ACL configuration:

Step 1 In the Exec Mode, enter the **show ip access-list** command.

The following is a sample output of this command. In this example, an ACL named *acl_1* was configured.

```
ip access list acl_1
  deny host 1.2.3.4
  deny ip any host 1.2.3.4
  permit any 1.2.4.4
1 ip access-lists are configured.
```

Applying IP ACLs

Once an ACL is configured, it must be applied to take effect.

Important: All ACLs should be configured and verified according to the instructions in the [Configuring ACLs on the System](#) section of this appendix prior to beginning these procedures. The procedures described below also assume that the subscribers have been previously configured.

As discussed earlier, you can apply an ACL to any of the following:

- [Applying an ACL to an Individual Interface](#)
- [Applying an ACL to All Traffic Within a Context](#) (known as a policy ACL)
- [Applying an ACL to an Individual Subscriber](#)
- [Applying a Single ACL to Multiple Subscribers](#)
- [Applying a Single ACL to Multiple Subscribers via APNs](#) (for 3GPP subscribers only)

Important: ACLs must be configured in the same context in which the subscribers and/or interfaces to which they are to be applied. Similarly, ACLs to be applied to a context must be configured in that context.

If ACLs are applied at multiple levels within a single context (such as an ACL is applied to an interface within the context and another ACL is applied to the entire context), they will be processed as shown in the following figure and table.

Figure 16. ACL Processing Order

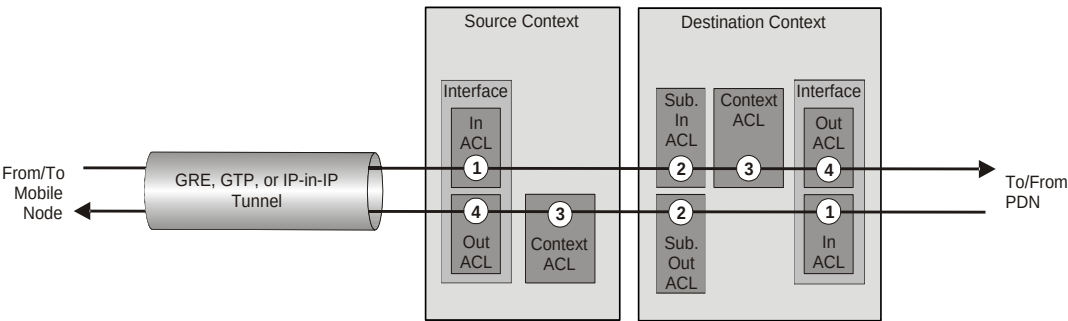


Table 51. ACL Processing Order Descriptions

Packet coming from the mobile node to the packet data network (left to right)	
Order	Description
1	An inbound ACL configured for the receiving interface in the Source Context is applied to the tunneled data (such as the outer IP header). The packet is then forwarded to the Destination Context.
2	An inbound ACL configured for the subscriber (either the specific subscriber or for any subscriber facilitated by the context) is applied.

3	A context ACL (policy ACL) configured in the Destination Context is applied prior to forwarding.
4	An outbound ACL configured on the interface in the Destination Context through which the packet is being forwarded is applied.
Packet coming from the packet data network to the mobile node (right to left)	
Order	Description
1	An inbound ACL configured for the receiving interface configured in the Destination Context is applied.
2	An outbound ACL configured for the subscriber (either the specific subscriber or for any subscriber facilitated by the context) is applied. The packet is then forwarded to the Source Context.
3	A context ACL (policy ACL) configured in the Source Context is applied prior to forwarding.
4	An outbound ACL configured on the interface in the Source Context through which the packet is being forwarded is applied to the tunneled data (i.e. the outer IP header).

In the event that an IP ACL is applied that has not been configured (for example, the name of the applied ACL was configured incorrectly), the system uses an “undefined” ACL mechanism for filtering the packet(s).

This section provides information and instructions for applying ACLs and for configuring an “undefined” ACL.

Applying an ACL to an Individual Interface

This section provides information and instructions for applying one or more ACLs to an individual interface configured on the system.



Important: This section provides the minimum instruction set for applying the ACL list to an interface on the system. For more information on commands that configure additional parameters and options, refer to the *Ethernet Interface Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

To configure the system to provide ACL facility to subscribers:

- Step 1** Apply the configured access control list by applying the example configuration in the [Applying ACL to Interface](#) section.
- Step 2** Verify that ACL is applied properly on interface by following the steps in the [Verifying the ACL Configuration on Interface](#) section.
- Step 3** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command save configuration. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Applying ACL to Interface

To apply the ACL to an interface, use the following configuration:

configure

```
context <acl_ctxt_name> [ -noconfirm ]

interface <interface_name>

ip access-group <acl_list_name> { in | out } [ <preference> ]

end
```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- The ACL to be applied must be configured in the context specified by this command.
- Up to 8 ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration on Interface

This section describes how to verify the ACL configuration.

Step 1 In the Exec Mode, enter the following command:

```
show configuration context context_name
```

context_name is the name of the context containing the interface to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure

context context_name

ip access-list acl_name

deny host ip_address

deny ip any host ip_address

exit

ip access-group access_group_name

service-redundancy-protocol

exit
```

```

interface interface_name

    ip address ip_address/mask

    exit

subscriber default

    exit

aaa group default

    exit

gtpv group default

    exit

end

```

Applying an ACL to All Traffic Within a Context

This section provides information and instructions for applying one or more ACLs to a context configured within a specific context on the system. The applied ACLs, known as policy ACLs, contain rules that apply to all traffic facilitated by the context.



Important: This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to the *Context Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

- Step 1** Apply the configured ACL as described in the [Applying ACL to Context](#) section.
- Step 2** Verify that ACL is applied properly on interface as described in the [Verifying the ACL Configuration in a Context](#) section.
- Step 3** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command `save configuration`. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Applying ACL to Context

To apply the ACLs to a context, use the following configuration:

configure

```

context <acl_ctxt_name> [ -noconfirm ]

    ip access-group <acl_list_name> [ in | out ] [ <preference> ]

end

```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- The context-level ACL are applied only to outgoing packets. The **in** and **out** keywords are deprecated and are only present for backward compatibility.
- The ACL to be applied must be configured in the context specified by this command.
- Up to 8 ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration in a Context

To verify the ACL configuration:

Step 1 Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
show configuration context context_name
```

context_name is the name of the context to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure

context context_name

    ip access-list acl_name

        deny host ip_address

        deny ip any host ip_address

    exit

    ip access-group access_group_name

    service-redundancy-protocol

    exit

    interface interface_name

        ip address ip_address/mask

    exit

    subscriber default

    exit

    aaa group default

    exit
```

```

gtpp group default

end

```

Applying an ACL to a RADIUS-based Subscriber

IP ACLs are applied to subscribers via attributes in their profile. The subscriber profile could be configured locally on the system or remotely on a RADIUS server.

To apply an ACL to a RADIUS-based subscriber, use the **Filter-Id** attribute. Refer to the *AAA and GTPP Interface Administration and Reference* for more detail on this attribute.

This section provides information and instructions for applying an ACL to an individual subscriber whose profile is configured locally on the system.

 **Important:** This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to the *Subscriber Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

- Step 1** Apply the configured access control list by applying the example configuration in the [Applying ACL to an Individual Subscriber](#) section.
- Step 2** Verify that ACL is applied properly on interface by following the steps in the [Verifying the ACL Configuration to an Individual Subscriber](#) section.
- Step 3** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command save configuration. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Applying an ACL to an Individual Subscriber

To apply the ACL to an individual subscriber, use the following configuration:

configure

```

context <acl_ctxt_name> [ -noconfirm ]

subscriber name <subs_name>

ip access-group <acl_list_name> [ in | out ]

end

```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- If neither the **in** nor the **out** keyword is specified, the ACL will be applied to all inbound and outbound packets.
- The ACL to be applied must be configured in the context specified by this command.

- Up to 8 ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration to an Individual Subscriber

These instructions are used to verify the ACL configuration.

Step 1 Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
show configuration context context_name
```

context_name is the name of the context containing the subscriber *subs1* to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure

context context_name

    ip access-list acl_name

        deny host ip_address

        deny ip any host ip_address

    exit

    ip access-group access_group_name

    service-redundancy-protocol

    exit

    interface interface

        ip address ip_address/mask

    exit

    subscriber default

    exit

    subscriber name subscriber_name

        ip access-group access_group_name in

        ip access-group access_group_name out

    exit

    aaa group default

    exit
```



```

gtpp group default

    exit

content-filtering server-group cfsg_name

    response-timeout response_timeout

    connection retry-timeout retry_timeout

end

```

Applying a Single ACL to Multiple Subscribers

As mentioned in the previous section, IP ACLs are applied to subscribers via attributes in their profile. The subscriber profile could be configured locally on the system or remotely on a RADIUS server.

The system provides for the configuration of subscriber functions that serve as default values when specific attributes are not contained in the individual subscriber's profile. The following table describes these functions.

Table 52. Functions Used to Provide “Default” Subscriber Attributes

Function	Description
Subscriber Named default	Within each context, the system creates a subscriber called default. The profile for the subscriber named default provides a configuration template of attribute values for subscribers authenticated in that context. Any subscriber attributes that are not included in a RADIUS-based subscriber profile is configured according to the values for those attributes as defined for the subscriber named default. NOTE: The profile for the subscriber named default is not used to provide missing information for subscribers configured locally.
default subscriber Command	This command in the PDSN, FA, and HA service Configuration modes specifies a profile from a subscriber named something other than default to use a configuration template of attribute values for subscribers authenticated in that context. This command allows multiple services to draw “default” subscriber information from multiple profiles.

When configured properly, the functions described in the table above could be used to apply an ACL to:

- All subscribers facilitated within a specific context by applying the ACL to the profile of the subscriber named *default*.
- All subscribers facilitated by specific services by applying the ACL to a subscriber profile and then using the **default subscriber** command to configure the service to use that subscriber as the “default” profile.

Applying an ACL to the Subscriber Named default

This section provides information and instructions for applying an ACL to the subscriber named *default*.



Important: This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to the *Subscriber Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

- Step 1** Apply the configured access control list by applying the example configuration in the [Applying an ACL to the Subscriber Named default](#) section.
- Step 2** Verify that ACL is applied properly on interface by following the steps in the [Verifying the ACL Configuration to the Subscriber Named default](#) section.
- Step 3** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command save configuration. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Applying an ACL to the Subscriber Named default

To example to apply the ACL to the default subscriber, use the following configuration:

configure

```
context <acl_ctxt_name> [ -noconfirm ]

  subscriber name <subs_name>

    ip access-group <acl_list_name> [ in | out ]

  end
```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- If neither the **in** nor the **out** keyword is specified, the ACL will be applied to all inbound and outbound packets.
- The ACL to be applied must be configured in the context specified by this command.
- Up to 8 ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration to the Subscriber Named default

These instructions are used to verify the ACL configuration.

- Step 1** Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
show configuration context context_name
```

context_name is the name of the context containing the subscriber default to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure

context context_name

    ip access-list acl_name

        deny host ip_address

        deny ip any host ip_address

    exit

ip access-group access_group_name

service-redundancy-protocol

    exit

interface interface

    ip address ip_address/mask

    exit

subscriber name default

    ip access-group access_group_name in

    ip access-group access_group_name out

    exit

aaa group default

    exit

gtpv group default

    exit

content-filtering server-group cfsq_name

    response-timeout response_timeout

    connection retry-timeout retry_timeout

end
```

Applying an ACL to Service-specified Default Subscribers

This section provides information and instructions for applying an ACL to the subscriber to be used as the “default” profile by various system services.



Important: This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer the *Subscriber Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

- Step 1** Apply the configured access control list by applying the example configuration in the [Applying an ACL to Service-specified Default Subscriber](#) section.
- Step 2** Verify that the ACL is applied properly on interface by following the steps in the [Verifying the ACL Configuration to Service-specified Default Subscriber](#) section.
- Step 3** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command save configuration. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Applying an ACL to Service-specified Default Subscriber

To apply the ACL to a service-specified Default subscriber, use the following configuration:

configure

```
context <acl_ctxt_name> [ -noconfirm ]

{ pdsn-service | fa-service | ha-service } <service_name>

  default subscriber <svc_default_subs_name>

  exit

subscriber name <svc_default_subs_name>

  ip access-group <acl_list_name> [ in | out ]

end
```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- If neither the **in** nor the **out** keyword is specified, the ACL will be applied to all inbound and outbound packets.
- The ACL to be applied must be configured in the context specified by this command.
- Up to 8 ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration to Service-specified Default Subscriber

To verify the ACL configuration.

Step 1 Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
show configuration context context_name
```

context_name is the name of the context containing the service with the default subscriber to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure

context context_name

    ip access-list acl_name

        deny host ip_address

        deny ip any host ip_address

    exit

ip access-group access_group_name

interface interface

    ip address ip_address/mask

    exit

subscriber default

    exit

subscriber name subscriber_name

    ip access-group access_group_name in

    ip access-group access_group_name out

    exit

pdsn-service service_name

    default subscriber subscriber_name

end
```

Applying a Single ACL to Multiple Subscribers via APNs

If IP ACLs are applied to subscribers via attributes in their profile, the subscriber profile could be configured locally on the system or remotely on a RADIUS server.

To reduce configuration time, ACLs can alternatively be applied to APN templates for GGSN subscribers. When configured, any subscriber packets facilitated by the APN template would then have the associated ACL applied.

This section provides information and instructions for applying an ACL to an APN template.



Important: This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to the *Subscriber Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

- Step 1** Apply the configured access control list by applying the example configuration in the [Applying an ACL to Multiple Subscriber via APNs](#) section.
- Step 2** Verify that ACL is applied properly on interface by following the steps in the [Verifying the ACL Configuration to APNs](#) section.
- Step 3** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command `save configuration`. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Applying an ACL to Multiple Subscriber via APNs

To apply the ACL to multiple subscribers via APN, use the following configuration:

configure

```
context <dest_context_name> [ -noconfirm ]

  apn <apn_name>

    ip access-group <acl_list_name> [ in | out ]

  end
```

Notes:

- The ACL to be applied must be in the destination context of the APN (which can be different from the context where the APN is configured).
- If either the **in** or **out** keyword is not specified, the command is added to the config file twice, once with **in** and once with **out**, and the ACL will be applied to all inbound and outbound packets.
- Up to 8 ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration to APNs

To verify the ACL configuration:

Step 1 Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
show configuration context context_name
```

context_name is the name of the context containing the APN *apn1* having *default* subscriber to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure

context context_name

    ip access-list acl_name

        deny host ip_address

        deny ip any host ip_address

    exit

ip access-group access_group_name

interface interface

    ip address ip_address/mask

    exit

subscriber default

    exit

apn apn_name

    ip access-group access_group_name in

    ip access-group access_group_name out

end
```


Appendix D

Congestion Control

This appendix describes the Congestion Control feature. It covers the following topics:

- [Overview](#)
- [Configuring Congestion Control](#)

Overview

Congestion Control monitors the system for conditions that could potentially degrade performance when the system is under heavy load. Typically, these conditions are temporary (for example, high CPU or memory utilization) and are quickly resolved. However, continuous or large numbers of these conditions within a specific time interval may impact the system's ability to service subscriber sessions. Congestion control helps identify such conditions and invokes policies for addressing the situation.

Congestion control operation is based on configuring the following:

- **Congestion Condition Thresholds:** Thresholds dictate the conditions for which congestion control is enabled and establishes limits for defining the state of the system (congested or clear). These thresholds function in a way similar to operation thresholds that are configured for the system as described in the *Thresholding Configuration Guide*. The primary difference is that when congestion thresholds are reached, a service congestion policy and an SNMP trap, `starCongestion`, are generated.

A threshold tolerance dictates the percentage under the configured threshold that must be reached in order for the condition to be cleared. An SNMP trap, `starCongestionClear`, is then triggered.

- **Port Utilization Thresholds:** If you set a port utilization threshold, when the average utilization of all ports in the system reaches the specified threshold, congestion control is enabled.
- **Port-specific Thresholds:** If you set port-specific thresholds, when any individual port-specific threshold is reached, congestion control is enabled system-wide.
- **Service Congestion Policies:** Congestion policies are configurable for each service. These policies dictate how services respond when the system detects that a congestion condition threshold has been crossed.



Important: This section provides the minimum instruction set for configuring congestion control. Commands that configure additional interface or port properties are provided in the *Subscriber Configuration Mode* chapter of the *Command Line Interface Reference*.

Configuring Congestion Control

To configure Congestion Control functionality:

- Step 1** Configure congestion control thresholds as described in the [Configuring the Congestion Control Threshold](#) section.
- Step 2** Configure service congestion policies as described in the [Configuring Service Congestion Policies](#) section.
- Step 3** Enable redirect overload policies as described in the [Enabling Congestion Control Redirect Overload Policy](#) section.
- Step 4** Configure disconnecting subscribers based on call or inactivity time as described in the [Disconnecting Subscribers Based on Call or Inactivity Time](#) section.
- Step 5** Save your configuration as described in the *Saving and Verifying Your Configuration* chapter in this guide.

Configuring the Congestion Control Threshold

To configure congestion control threshold, apply the following example configuration in the Global Configuration mode of the CLI:

```
configure

  congestion-control threshold max-sessions-per-service-utilization <percent>

  congestion-control threshold tolerance <percent>

end
```

Notes:

- There are several additional threshold parameters. See the *Global Configuration Mode* chapter of the *Command Line Interface Reference* for more information.
- The tolerance is the percentage under a configured threshold that dictates the point at which the condition is cleared.
- Repeat this configuration as needed for additional thresholds.

Configuring Service Congestion Policies

To create a congestion control policy, apply the following example configuration in the Global Configuration mode of the CLI:

```
configure

  congestion-control policy <service> action { drop | none | redirect | reject }

end
```

Notes:

- When the redirect action occurs for PDSN services, the PDSN responds to the PCF with a reply code of 136, “unknown PDSN address” along with the IP address of an alternate PDSN.
- **redirect** is not available for PDIF.
- The default action for PDIF is “none.”
- When the redirect action occurs for HA services, the system responds to the FA with a reply code of 136, “unknown home agent address”.
- **redirect** can not be used in conjunction with GGSN services.
- **redirect** is not available for the LMA service.
- When setting the action to **reject**, the reply code is 130, “insufficient resources”.
- For the GGSN, the reply code is 199, “no resources available”.
- For the MME, **redirect** is not available.

Configuring Overload Reporting on the MME

When an overload condition is detected on an MME and the report-overload keyword is enabled in the **congestion-control policy** command, the MME reports the condition to a specified percentage of eNodeBs and proceeds to take the configured action on incoming sessions. To create a congestion control policy with overload reporting, apply the following example configuration:

```
configure

  congestion-control policy mme-service action report-overload reject-new-sessions
  enodeb-percentage <percentage>

end
```

Notes:

- Other overload actions include **permit-emergency-sessions** and **reject-non-emergency-sessions**.

Enabling Congestion Control Redirect Overload Policy

To create a congestion control policy and configure a redirect overload policy for the service, apply the following example configuration:

 **Important:** Redirect is not available on PDIF for this release.

```
configure

  congestion-control

  context <context_name>

    {service_configuration_mode}

    policy overload redirect address
```

```
end
```

Notes:

- *Optional:* If the congestion control policy action was configured to **redirect**, then a redirect overload policy must be configured for the service(s) that are affected.
- There are several service configuration modes that you can configure. See the *Command Line Interface Reference* for a complete list of modes.
- You can set various options for redirection. See the *Command Line Interface Reference* for more information.
- Repeat this configuration example to configure overload policies for additional services configured in the same context.

Verify the Service Overload Policies

To verify that the service overload policies were properly configured enter the following command in the Exec Mode:

```
show service_type name service_name
```

This command lists the entire service configuration. Verify that the information displayed for the “Overload Policy” is accurate.

Repeat this configuration example to configure additional services in other contexts.

Verify the Congestion Control Configuration

To verify Congestion Control Configuration enter the **show congestion-control configuration** command in the Exec Mode.

The following output is a concise listing of all threshold and policy configurations:

```
Congestion-control: enabled

Congestion-control threshold parameters

system cpu utilization: 80%

service control cpu utilization: 80%

system memory utilization: 80%

message queue utilization: 80%

message queue wait time: 10 seconds

port rx utilization: 80%

port tx utilization: 80%

license utilization: 100%

max-session-per-service utilization: 100%

tolerance limit: 10%

Overload-disconnect: disabled
```

Overload-disconnect threshold parameters

```
license utilization: 80%  
max-session-per-service utilization: 80%  
tolerance: 10%  
session disconnect percent: 5%  
iterations-per-stage: 8
```

Congestion-control Policy

```
pdsn-service: none  
ha-service: none  
lma-service: none  
ggsn-service: none  
closedrp-service: none  
lns-service: none  
cscf-service: reject  
pdif-service: none  
fng-service: none  
sgsn-service: none  
mme-service: drop  
asngw-service: none  
asnpc-service: none  
phsgw-service: none  
phspc-service: none  
mipv6ha-service: none  
lma-service: none  
sgw-service: none  
pgw-service: none  
hnbgw-service: none  
pcc-policy-service: none  
pcc-quota-service: none
```

```
pcc-af-service: none
```

The primary threshold to observe is *license utilization*. This threshold is defaulted to 80%. Overload controls on the system enables the Congestion-control Policy when the system has only 80% of the licenses used. The overload condition will not clear until the utilization drops below the tolerance limit setting. The tolerance limit is defaulted to 10%. If the system goes into overload due to license utilization (threshold at 80%), the overload condition will not clear until the license utilization reaches 70%.

The system may go into overload if threshold settings are set too low and congestion control is enabled. You will need to review all threshold values and become familiar with the settings.

Since the recommendation for license utilization overload threshold is 100%, you should enable a license threshold alarm at 80%. An alarm is then triggered when the license utilization hits 80%. When the congestion-control policy setting is set to **drop**, the system drops incoming packets containing new session requests.



Important: For additional information on configuring the alarm threshold, refer to the *Threshold Configuration Guide*.

Disconnecting Subscribers Based on Call or Inactivity Time

During periods of heavy system load, it may be necessary to disconnect subscribers in order to maintain an acceptable level of system performance. You can establish thresholds to select subscribers to disconnect based on the length of time that a call has been connected or inactive.

To enable overload disconnect for the currently selected subscriber, use the following configuration example:

```
configure

context <context_name>

    subscriber name <subscriber_name>

        default overload-disconnect threshold inactivity-time <dur_thresh>

        default overload-disconnect threshold connect-time <dur_thresh>

    end
```

To disable the overload disconnect feature for this subscriber, use the following configuration example:

```
configure

context <context_name>

    subscriber <subscriber_name>

        no overload-disconnect {[threshold inactivity-time] | [threshold connect-time]}

    end
```

Notes:

- **overload-disconnect** is not supported for the CSCF service.

Appendix E

Content Service Steering

This chapter provides information on configuring Content Service Steering (CSS). The product administration guides provide provides examples and procedures for configuration of basic services on the system. You should select the configuration example that best meets your service model, and configure the required elements for that model as described in the respective product administration guide, before using the procedures in this appendix.



Important: Internal CSS is a generic feature, if an ECSv2 license is installed on your system, internal CSS can be enabled. A separate license is not required to enable internal CSS. Contact your local Cisco account representative for information on how to obtain a license.

This chapter contains the following topics:

- [Overview](#)
- [Configuring Internal Content Service Steering](#)

Overview

Content Service Steering (CSS) selectively directs subscriber traffic to In-line services internal to the system based on data content presented by mobile subscribers. CSS is a broad term that includes features such as NAT, HTTP redirection, and DNS redirection.

CSS uses Access Control Lists (ACLs) to redirect subscriber traffic flows. ACLs control the flow of packets into and out of the system. ACLs consist of “rules” (ACL rules) or filters that control the action taken on packets matching the filter criteria.

ACLs are configurable on a per-context basis and applies to a subscriber through either a subscriber profile (or an APN profile in the destination context. For additional information, refer to the *Access Control Lists* appendix in this guide

Configuring Internal Content Service Steering

To configure and activate a single CSS service for redirecting all of a subscriber's IP traffic to an internal in-line service:

- Step 1** Define an IP ACL as described in the [Defining IP Access Lists for Internal CSS](#) section.
- Step 2** *Optional:* Apply an ACL to an individual subscriber as described in the [Applying an ACL to an Individual Subscriber \(Optional\)](#) section.
- Step 3** *Optional:* Apply a single ACL to multiple subscribers as described in the [Applying an ACL to Multiple Subscribers \(Optional\)](#) section.
- Step 4** *Optional:* Apply an ACL to multiple subscribers via APNs as described in the [Applying an ACL to Multiple Subscribers via APNs \(Optional\)](#) section.
- Step 5** Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command save configuration. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.



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. Not all commands or keywords/variables may be supported or available. Availability varies on the platform type and installed license(s).

Defining IP Access Lists for Internal CSS

IP ACLs specify what type of subscriber traffic and which direction (uplink, downlink, or both) traffic is redirected. The IP ACL must be specified in the context in which subscriber authentication is performed.



Caution: To minimize the risk of data loss, do not make configuration changes to ACLs while the system is facilitating subscriber sessions.

Use the following configuration example to define an IP ACL for internal CSS; start in the Exec mode of the CLI:

configure

```
context <context_name>

  ip access-list <acl_name>

    redirect css service <service_name> <keywords> <options>

  end
```

Notes:

- <service_name> must be an ACL service name.

- For information on the keywords and options available with the **redirect css service** command, see the *ACL Configuration Mode Commands* chapter of the *Command Line Interface Reference*.
- For IPv6 ACLs, the same configurations must be done in the IPv6 ACL Configuration Mode. See the *IPv6 ACL Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Applying an ACL to an Individual Subscriber (Optional)

For information on how to apply an ACL to an individual subscriber, refer to the *Applying an ACL to an Individual Subscriber* section of the *Access Control Lists* appendix.

Applying an ACL to Multiple Subscribers (Optional)

IP ACLs are applied to subscribers via attributes in their profiles. The subscriber profile can be configured locally on the system or remotely on a RADIUS server.

The system provides for the configuration of subscriber functions that serve as default values when specific attributes are not contained in the individual subscriber's profile. When configured properly, the functions can be used to apply an ACL to:

- All subscribers facilitated within a specific context by applying the ACL to the profile of the subscriber named *default*.
- All subscribers facilitated by specific services by applying the ACL to a subscriber profile and then using the **default subscriber** command to configure the service to use that subscriber as the “default” profile.

Applying an ACL to the Subscriber Named default (Optional)

For information on how to apply an ACL to the default subscriber, refer to the *Applying an ACL to the Subscriber Named default* section of the *Access Control Lists* appendix.

Applying an ACL to Service-specified Default Subscribers (Optional)

For information on how to apply an ACL to the subscriber to be used as the “default” profile by various system services, refer to the *Applying an ACL to Service-specified Default Subscribers* section of the *Access Control Lists* appendix.

Applying an ACL to Multiple Subscribers via APNs (Optional)

IP ACLs are applied to subscribers via attributes in their profiles. The subscriber profile can be configured locally on the system or remotely on a RADIUS server.

To reduce configuration time, ACLs can alternatively be applied to APN templates. When configured, any subscriber packets facilitated by the APN template would then have the associated ACL applied.

For information on how to apply an ACL to multiple subscribers via APNs, refer to the *Applying a Single ACL to Multiple Subscribers via APNs* section the *Access Control Lists* chapter.

Appendix F

Interchassis Session Recovery

This appendix describes how to configure Interchassis Session Recovery (ICSR). The product Administration Guides provide examples and procedures for configuration of basic services on the system. You should select the configuration example that best meets your service model, and configure the required elements for that model as described in the respective product Administration Guide, before using the procedures in this appendix.



Important: ICSR is a licensed Cisco feature that requires a separate license. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of the *Software Management Operations* chapter.

This appendix discusses the following:

- [Overview](#)
- [ICSR Operation](#)
- [Configuring Interchassis Session Recovery \(ICSR\)](#)



Caution: ICSR should not be configured on chassis supporting L2TP calls.

Overview

The ICSR feature provides the highest possible availability for continuous call processing without interrupting subscriber services. ICSR allows the operator to configure geographically distant gateways for redundancy purposes. In the event of a node or gateway failure, ICSR allows sessions to be transparently routed around the failure, thus maintaining the user experience. ICSR also preserves session information and state.

ICSR is implemented through the use of redundant chassis. The chassis are configured as primary and backup, with one being active and one standby. Both chassis are connected to the same AAA server. A checkpoint duration timer controls when subscriber data is sent from the active chassis to the standby chassis. If the active chassis handling the call traffic goes out of service, the standby chassis transitions to the active state and continues processing the call traffic without interrupting the subscriber session.

The chassis determine which is active through a proprietary TCP-based connection known as the Service Redundancy Protocol (SRP) link. The SRP link is used to exchange Hello messages between the primary and backup chassis and must be maintained for proper system operation.



Important: Refer to the *Product Overview* to verify whether a specific service supports ICSR as an option.

Interchassis Communication

Chassis configured to support ICSR communicate using periodic Hello messages. These messages are sent by each chassis to notify the peer of its current state. The Hello message contains information about the chassis such as its configuration and priority. A dead interval is used to set a time limit for a Hello message to be received from the chassis' peer. If the standby chassis does not receive an Hello message from the active chassis within the dead interval, the standby chassis transitions to the active state. In situations where the SRP link goes out of service, a priority scheme is used to determine which chassis processes the session. The following priority scheme is used:

- route modifier
- chassis priority
- SPIO MAC address

Checkpoint Messages

Checkpoint messages are sent from the active chassis to the standby chassis. These messages are sent at specific intervals and contain all the information needed to recreate the sessions on the standby chassis, if that chassis were to become active. Once a session exceeds the checkpoint duration, checkpoint data is collected on the session.

AAA Monitor

AAA servers are monitored using the authentication probe mechanism. AAA servers are considered Up if the authentication-probe receives a valid response. AAA servers are considered Down when the **max-retries count** specified in the configuration of the AAA server has been reached. SRP will initiate a switchover when none of the configured AAA servers responds to an authentication probe. AAA probing is only performed on the active chassis.



Important: A switchover event caused by an AAA monitoring failure is non-reversible.

If the newly active chassis fails to monitor the configured AAA servers, it remains as the active chassis until one of the following occurs:

- a manual switchover
- another non-AAA failure event causes the system to switchover
- a CLI command is used to clear the AAA failure flag and allow the chassis to switch to standby

BGP Interaction

The Service Redundancy Protocol implements non-reversible switchover behavior via a mechanism that adjusts the route modifier value for the advertised loopback/IP Pool routes. The initial value of the route modifier value is determined by the chassis' configured role and is initialized to a value that is higher than a normal operational value. This ensures that in the event of an SRP link failure and an SRP task failure, the correct chassis is still preferred in the routing domain.

The Active and Standby chassis share current route modifier values. When BGP advertises the loopback and IP pool routes, it converts the route modifier into an autonomous systems (AS) path prepend count. The Active chassis always has a lower route modifier, and thus prepends less to the AS-path attribute. This causes the route to be preferred in the routing domain.

If communication on the SRP link is lost, and both chassis in the redundant pair are claiming to be Active, the previously Active chassis is still preferred since it is advertising a smaller AS-path into the BGP routing domain. The route modifier is incremented as switchover events occur. A threshold determines when the route modifier should be reset to its initial value to avoid rollover.

Requirements

ICSR configurations require the following:

- Two chassis configured for the same service types. The services must be bound on an SRP-activated loopback interface.
- Both chassis must have identical hardware.
- Three contexts:
 - **Redundancy** – to configure the primary and backup chassis redundancy.
 - **Source** – AAA configuration of the specified nas-ip-address must be the IP address of an interface bound to an HA, or any core network service configured within the same context.
 - **Destination** – to configure monitoring and routing to the PDN.
- AAA RADIUS server
- Border Gateway Protocol (BGP) – ICSR uses the route modifier to determine the chassis priority.

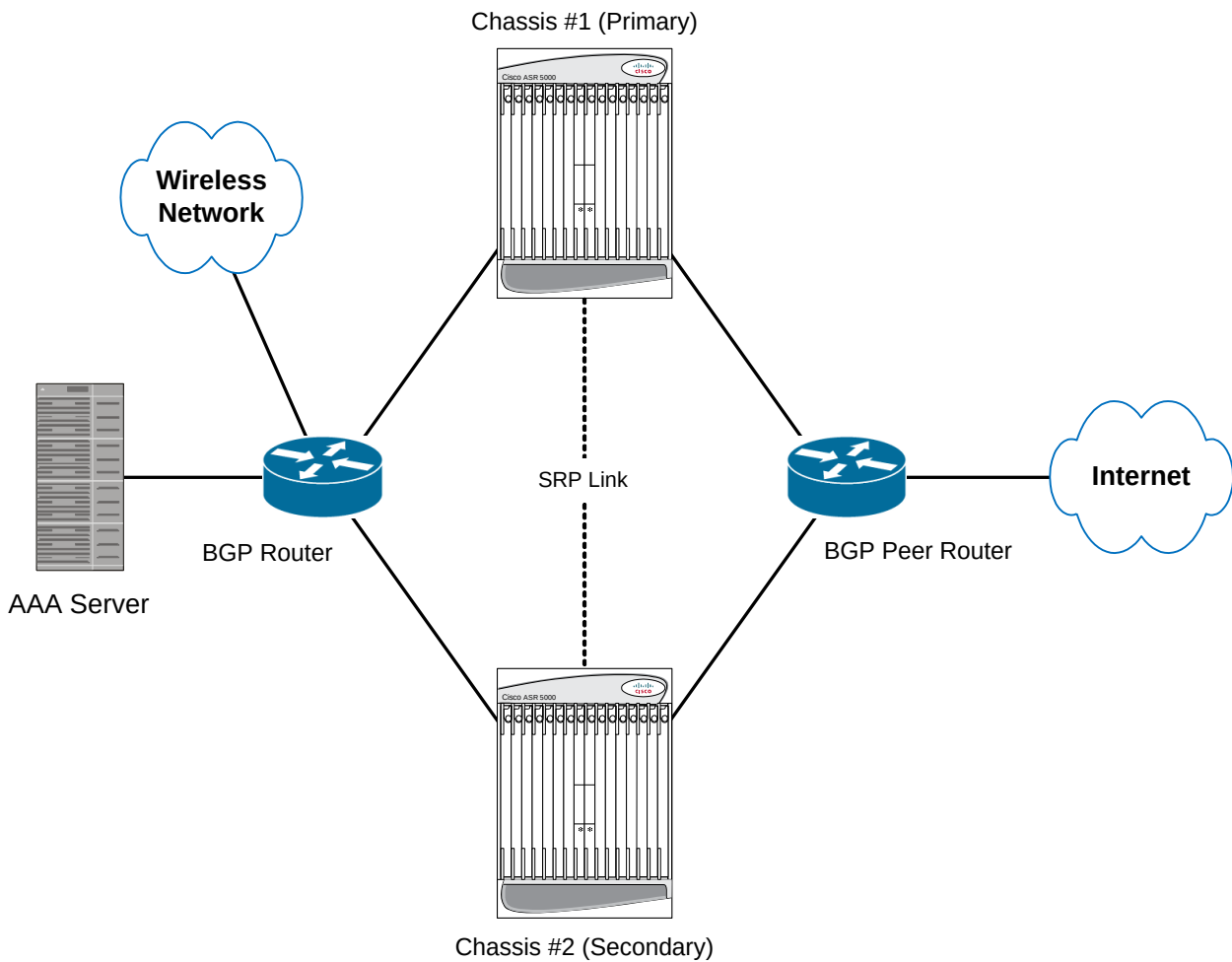


Important: ICSR is a licensed Cisco feature. Verify that each chassis has the appropriate license before using the procedures in this appendix. To do this, log in to both chassis and execute a **show license information** command. Look for "Inter-Chassis Session Recovery". If the chassis is not licensed, please contact your Cisco account representative.

Caution: ICSR should not be configured for chassis supporting L2TP calls.

The following figure shows an ICSR network.

Figure 17. ASR 5000 ICSR Network

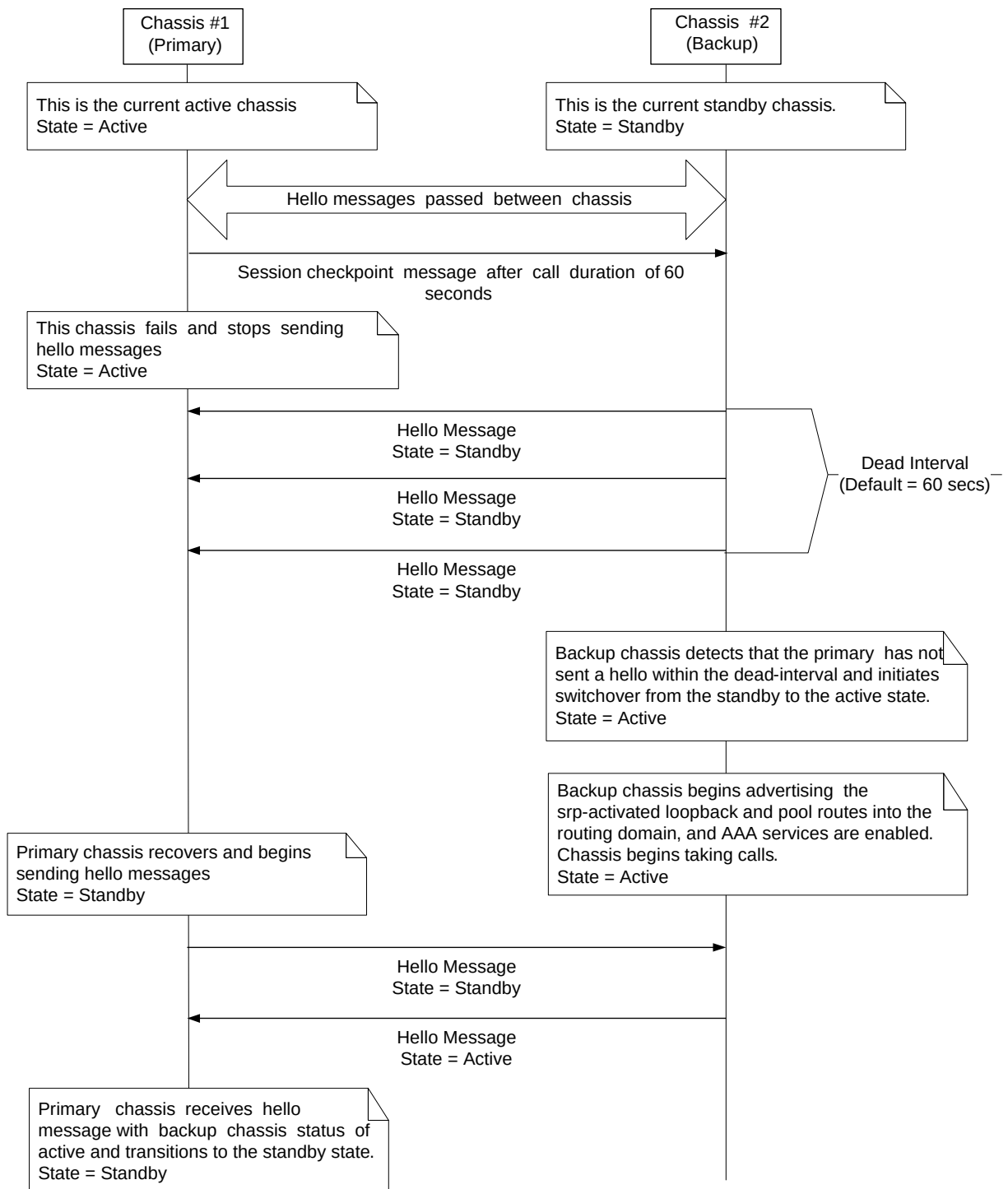


ICSR Operation

This section shows operational flows for ICSR.

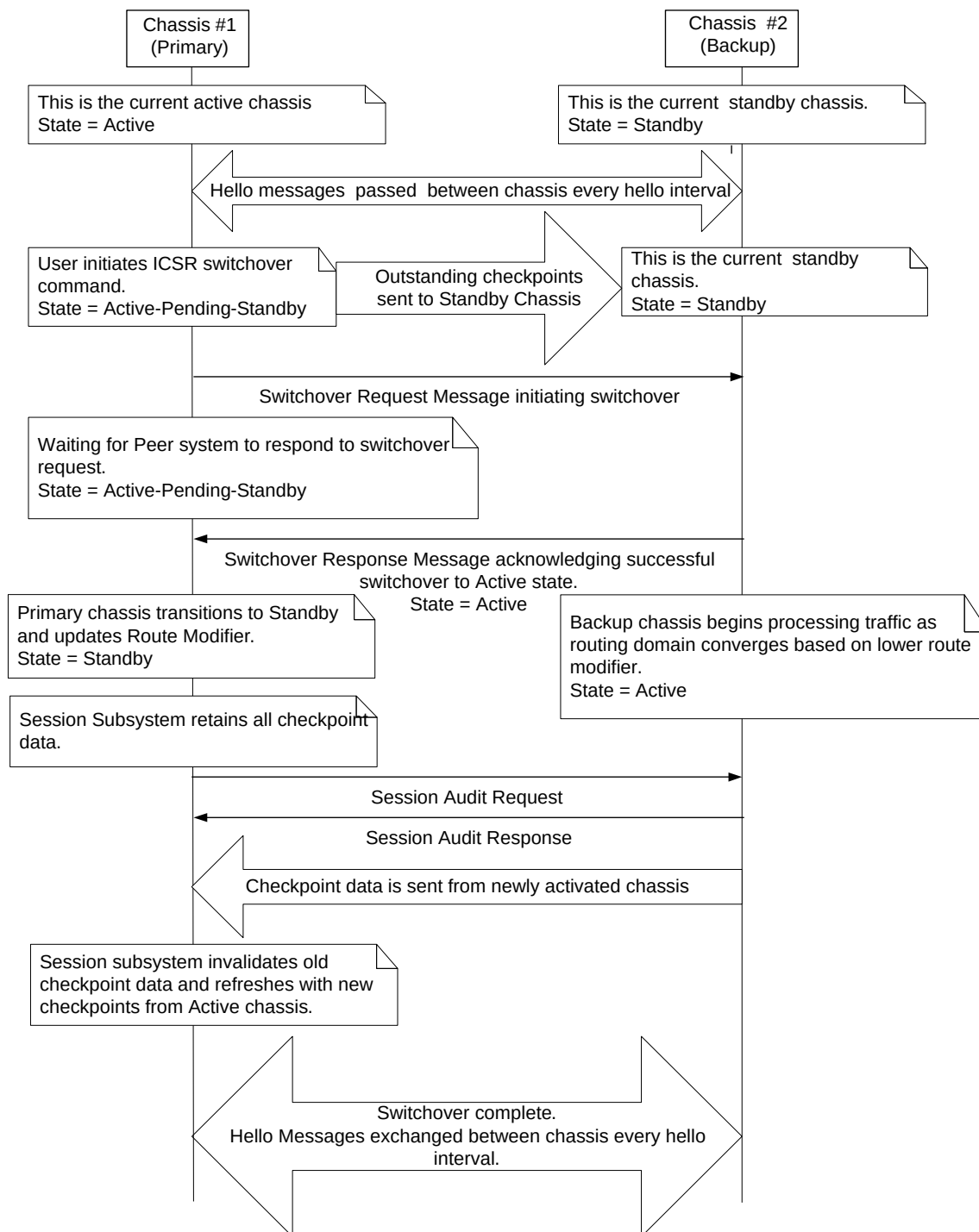
The following figure shows an ICSR process flow due to a primary failure.

Figure 18. ICSR Process Flow (Primary Failure)



The following figure shows an ICSR process flow due to a manual switchover.

Figure 19. ICSR Process Flow (Manual Switchover)



Chassis Initialization

When the chassis are simultaneously initialized, they send Hello messages to their configured peer. The peer sends a response, establishes communication between the chassis, and messages are sent that contain configuration information.

During initialization, if both chassis are misconfigured in the same mode - both active (primary) or both standby (backup), the chassis with the highest priority (highest number set with the ICSR **priority** command) becomes active and the other chassis becomes the standby.

If the chassis priorities are the same, the system compares the two MAC addresses and the chassis with the higher SPIO MAC address becomes active. For example, if the chassis have MAC addresses of *00-02-43-03-1C-2B* and *00-02-43-03-01-3B*, the last 3 sets of octets (the first 3 sets are the vendor code) are compared. In this example, the *03-1C-2B* and *03-01-3B* are compared from left to right. The first pair of octets in both MAC addresses are the same, so the next pairs are compared. Since the *01* is lower than the *1C*, the chassis with the SPIO MAC address of *00-02-43-03-1C-2B* becomes active and the other chassis the standby.

Chassis Operation

This section describes how the chassis communicate, maintain subscriber sessions, and perform chassis switchover.

Chassis Communication

If one chassis is in the active state and one in the standby state, they both send Hello messages at each hello interval. Subscriber sessions that exceed the checkpoint session duration are included in checkpoint messages that are sent to the standby chassis. The checkpoint message contains subscriber session information so if the active chassis goes out of service, the backup chassis becomes active and is able to continue processing the subscriber sessions. Additional checkpoint messages occur at various intervals whenever subscriber session information is updated on the standby chassis.

Chassis Switchover

If the active chassis goes out of service, the standby chassis continues to send Hello messages. If the standby chassis does not receive a response to the Hello messages within the dead interval, the standby chassis initiates a switchover. During the switchover, the standby chassis begins advertising its srp-activated loopback and pool routes into the routing domain. Once the chassis becomes active, it continues to process existing AAA services and subscriber sessions that had checkpoint information, and is also able to establish new subscriber sessions.

When the primary chassis is back in service, it sends Hello messages to the configured peer. The peer sends a response, establishes communication between the chassis, and sends Hello messages that contain configuration information. The primary chassis receives an Hello message that shows the backup chassis state as active and then transitions to standby. The Hello messages continue to be sent to each peer, and checkpoint information is now sent from the active chassis to the standby chassis at regular intervals.

When chassis switchover occurs, the session timers are recovered. The access gateway session recovery is recreated with the full lifetime to avoid potential loss of the session and the possibility that a renewal update was lost in the transitional checkpoint update process.

Configuring Interchassis Session Recovery (ICSR)



Important: The ICSR configuration must be the same on the primary and backup chassis. If each chassis has a different Service Redundancy Protocol (SRP) configuration, the session recovery feature does not function and sessions cannot be recovered when the active chassis goes out of service.

This section describes how to configure basic ICSR on each chassis. For information on commands that configure additional parameters and options, refer to the *Command Line Interface Reference*.



Caution: ICSR should not be configured for chassis supporting L2TP calls.

The procedures described below assume the following:

- The chassis have been installed and configured with core network services.
For more configuration information and instructions on configuring services, refer to the respective product Administration Guide.
- In addition, the IP address pools must be **srp activated**.
- AAA server is installed, configured and accessible by both chassis.
For more information on configuring the AAA server, refer to the *AAA Interface Administration and Reference*.
- BGP router installed and configured. See the *Routing* appendix in this guide for more information on configuring BGP services.

To configure the ICSR on a primary and/or backup chassis:

- Step 1** Configure the SRP context by applying the example configuration in the [Configuring the Service Redundancy Protocol \(SRP\) Context](#) section.
- Step 2** Modify the source context of the core network service by applying the example configuration in the [Modifying the Source Context for ICSR](#) section.
- Step 3** Modify the destination context of core network service by applying the example configuration in the [Modifying the Destination Context for ICSR](#) section.
- Step 4** *Optional:* Disable bulk statistics collection on the standby system by applying the example configuration in the [Disabling Bulk Statistics Collection on a Standby System](#) section.
- Step 5** Verify your primary and backup chassis configuration by following the steps in the [Verifying the Primary and Backup Chassis Configuration](#) section.
- Step 6** Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

Configuring the Service Redundancy Protocol (SRP) Context

To configure the system to work with ICSR:

- Step 1** Create the chassis redundancy context and bind it to the IP address of the primary chassis by applying the example configuration in the [Creating and Binding the SRP Context](#) section.
- Step 2** Configure the chassis redundancy context with priority, chassis mode, hello interval, dead-interval and peer IP address by applying the example configuration in the [Configuring the SRP Context Parameters](#) section.
- Step 3** Configure the SRP context with interface parameters (including interface name, IP address and port number) for interchassis communication by applying the example configuration in the [Configuring the SRP Context Interface Parameters](#) section.
- Step 4** Verify your SRP context configuration by following the steps in the [Verifying SRP Configuration](#) section.
- Step 5** Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

Creating and Binding the SRP Context

Use the example below to create the SRP context and bind it to primary chassis IP address:



Important: ICSR is configured using two systems. Be sure to create the redundancy context on both systems. CLI commands must be executed on both systems. Log onto both chassis before continuing. Always make configuration changes on the primary chassis first. Before starting this configuration, identify which chassis to configure as the primary and use that login session.

```
configure

context <srp_ctxt_name> [ -noconfirm ]

    service-redundancy-protocol

        bind address <ip_address>

    end
```

Notes:

- ICSR should be configured and maintained in a separate context.
- Be sure to bind the local IP address to the primary chassis. When configuring the backup chassis, be sure to bind the local IP address to the backup chassis.

Configuring the SRP Context Parameters

This configuration assigns a chassis mode and priority, and also configures the redundancy link between the primary and backup chassis:



Important: CLI commands must be executed on both chassis. Log onto both chassis before continuing. Always make configuration changes on the primary chassis first.

configure

```
context <srp_ctxt_name>

    service-redundancy-protocol

        chassis-mode { primary | backup }

        priority <priority>

        peer-ip-address <ip_address>

        hello-interval <dur_sec>

        dead-interval <dead_dur_sec>

    end
```

Notes:

- ICSR should be configured and maintained in a separate context.
- When assigning the chassis mode on the backup chassis be sure to enter the **backup** keyword.
- The **priority** determines which chassis becomes active when the redundancy link goes out of service. The higher priority chassis has the lower number. Be sure to assign different priorities to each chassis.
- Enter the IP chassis of the backup chassis as the **peer-ip-address** to the primary chassis. Assign the IP address of the primary chassis as the **peer-ip-address** to the backup chassis.
- The **dead-interval** must be at least three times greater than the **hello-interval**. For example, if the hello interval is 10, the dead interval should be at least 30. System performance is severely impacted if the hello interval and dead interval are not set properly.

Configuring the SRP Context Interface Parameters

This procedure configures the communication interface with the IP address and port number within the SRP context. This interface supports interchassis communication.



Important: CLI commands must be executed on both chassis. Log onto both chassis before continuing. Always make configuration changes on the primary chassis first.

configure

```
context <vpn_ctxt_name> [ -noconfirm ]

    interface <srp_if_name>

        ip-address { <ip_address> | <ip_address>/<mask> }

    exit
```

```

    exit

port ethernet <slot_num>/<port_num>

    description <des_string>

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

    no shutdown

bind interface <srp_if_name> <srp_ctxt_name>

end

```

Verifying SRP Configuration

Step 1 Verify that your SRP contexts were created and configured properly by entering the **show srp info** command (Exec Mode).

Sample output for this command as shown. In this example, an SRP context called *srp1* was configured with default parameters.

```

Service Redundancy Protocol:
-----

Context: srp1

Local Address: 0.0.0.0

Chassis State: Init

Chassis Mode: Backup

Chassis Priority: 125

Local Tiebreaker: 00-00-00-00-00-00

Route-Modifier: 34

Peer Remote Address: 0.0.0.0

Peer State: Init

Peer Mode: Init

Peer Priority: 0

Peer Tiebreaker: 00-00-00-00-00-00

Peer Route-Modifier: 0

Last Hello Message received: -

Peer Configuration Validation: Initial

```



```

Last Peer Configuration Error: None

Last Peer Configuration Event: -

Connection State: None

```

Modifying the Source Context for ICSR

To modify the source context of core service:

- Step 1** Add the Border Gateway Protocol (BGP) router AS-path and configure the gateway IP address, neighbor IP address, remote IP address in the source context where the core network service is configured, by applying the example configuration in the [Configuring BGP Router and Gateway Address](#) section.
- Step 2** Configure the service redundancy context with the BGP neighbor context and IP address to monitor the BGP link activity by applying the example configuration in the [Configuring SRP Context for BGP](#) section.
- Step 3** Verify your BGP context configuration by following the steps in the [Verifying BGP Configuration](#) section.
- Step 4** Save your configuration as described in the *Verifying and Saving Your Configuration* chapter in this guide.

Configuring BGP Router and Gateway Address

Use the following example to create the BGP context and network addresses.

configure

```

context <source_ctxt_name>

  router bgp <AS_num>

    network <gw_ip_address>

    neighbor <neighbor_ip_address> remote-as <AS_num>

  end

```

Notes:

- *source_ctxt_name* is the context where the core network service is configured.

Configuring SRP Context for BGP

Use the following example to configure the BGP context and IP addresses in the SRP context.

configure

```

context <srp_ctxt_name>

  service-redundancy-protocol

    monitor bgp context <source_ctxt_name> <neighbor_ip_address>

```

```
end
```

Verifying BGP Configuration

Step 1 Verify your BGP configuration by entering the **show srp monitor bgp** command (Exec Mode).

Modifying the Destination Context for ICSR

To modify the destination context of core service:

- Step 1** Add the BGP router and configure the gateway IP address, neighbor IP address, remote IP address in the destination context where the core network service is configured, by applying the example configuration in the [Configuring BGP Router and Gateway Address in Destination Context](#) section.
- Step 2** Configure the service redundancy context with BGP neighbor context and IP address to monitor the BGP link activity by applying the example configuration in the [Configuring SRP Context for BGP for Destination Context](#) section.
- Step 3** Set the subscriber mode to **default** by following the steps in the [Setting Subscriber to Default Mode](#) section.
- Step 4** Verify your BGP context configuration by following the steps in the [Verifying BGP Configuration in Destination Context](#) section.
- Step 5** Save your configuration as described in the *Verifying and Saving Your Configuration* chapter in this guide.

Configuring BGP Router and Gateway Address in Destination Context

Use the following example to create the BGP context and network addresses.

```
configure

context <dest_ctxt_name>

    router bgp <AS_num>

        network <gw_ip_address>

        neighbor <neighbor_ip_address> remote-as <AS_num>

end
```

Notes:

- *AS_num* is the autonomous systems path number for this BGP router.

Configuring SRP Context for BGP for Destination Context

Use the following example to configure the BGP context and IP addresses in the SRP context.

```
configure

context <srp_ctxt_name>
```

```
service-redundancy-protocol

  monitor bgp context <dest_ctxt_name> <neighbor_ip_address>

end
```

Setting Subscriber to Default Mode

Use the following example to set the subscriber mode to **default**.

```
configure

context <dest_ctxt_name>

  subscriber default

end
```

Verifying BGP Configuration in Destination Context

Step 1 Verify your BGP configuration by entering the **show srp monitor bgp** command (Exec Mode).

Disabling Bulk Statistics Collection on a Standby System

You can disable the collection of bulk statistics from a system when it is in the standby mode of operation.

 **Important:** When this feature is enabled and a system transitions to standby state, any pending accumulated statistical data is transferred at the first opportunity. After that no additional statistics gathering takes place until the system comes out of standby state.

Use the following example to disable the bulk statistics collection on a standby system.

```
configure

  bulkstat mode

    no gather-on-standby

  end
```

Repeat this procedure for both systems.

Verifying the Primary and Backup Chassis Configuration

This section describes how to compare the ICSR configuration on both chassis.

Step 1 Enter the **show configuration srp** command on both chassis (Exec mode).

Verify that both chassis have the same SRP configuration information. The output looks similar to following:

```
config

context source

interface haservice loopback

ip address 172.17.1.1 255.255.255.255 srp-activate

#exit

radius attribute nas-ip-address address 172.17.1.1

radius server 192.168.83.2 encrypted key 01abd002c82b4a2c port 1812

radius accounting server 192.168.83.2 encrypted key 01abd002c82b4a2c port 1813

ha-service ha-pdsn

mn-ha-spi spi-number 256 encrypted secret 6c93f7960b726b6f6c93f7960b726b6f hash-
algorithm md5

fa-ha-spi remote-address 192.168.82.0/24 spi-number 256 encrypted secret
1088bdd6817f64df

bind address 172.17.1.1

#exit

#exit

context destination

ip pool dynamic 172.18.0.0 255.255.0.0 public 0 srp-activate

ip pool static 172.19.0.0 255.255.240.0 static srp-activate

#exit

context srp

service-redundancy-protocol

#exit

#exit

end
```

Appendix G

QoS Management

This appendix describes the Quality of Service (QoS) management on Cisco® ASR 5000 chassis and explains how it is configured.

The product Administration Guides provide examples and procedures for configuration of basic services on the system. You should select the configuration example that best meets your service model and configure the required elements for that model as described in the respective product Administration Guide, before using the procedures in this appendix.

This appendix describes the following topics:

- [Introduction](#)
- [Dynamic QoS Renegotiation](#)
- [Network Controlled QoS \(NCQoS\)](#)
- [Configuring Dynamic QoS Renegotiation](#)
- [Configuring Network Controlled QoS \(NCQoS\)](#)
- [Monitoring Dynamic QoS Renegotiation Operation](#)

Introduction

The QoS Traffic Policing functionality supported by the GGSN implements QoS for subscribers based on the configuration of the APN template. As a result, all subscriber PDP contexts using the APN receive the same QoS level. This could lead to unused or under-utilized bandwidth by some subscribers thus reducing the amount of resources available to others.

Dynamic QoS Renegotiation

Dynamic QoS Renegotiation minimizes the risk of bandwidth mis-appropriation. This feature allows the GGSN to analyze application traffic, and trigger QoS renegotiation with the SGSN to optimize service performance.

In Dynamic QoS Renegotiation, the GGSN performs packet inspection of application traffic to detect the type of service being utilized and automatically renegotiates the QoS to the appropriate level with a maximum QoS level corresponding to the level granted by the HLR.

QoS renegotiation is performed by sending an Update PDP Context Request to the SGSN. This solution is optimal since the appropriate QoS level is always granted to the subscriber without any requirement on the handset or on the core network. The only prerequisite is QoS renegotiation support on the SGSN. In this model, over reservation of radio resources is avoided, while maintaining the appropriate bandwidth for subscribers with real requirements.

The ASR 5000 supports L7 stateful analysis and QoS Renegotiation. These functions combine to become Dynamic QoS Renegotiation. The system also generates CDRs (or real time charging information) that includes the current QoS information and the service accessed. This enables intelligent application-based charging of services, taking into account the granted QoS. It also enables rebates when it was not possible to provide the QoS level required by an application.



Important: For L7 traffic analysis an ECSv2 license is required.

How Dynamic QoS Renegotiation Works

Implementation of Dynamic QoS Renegotiation involves the following:

- Initial QoS
- Service Detection
- Classification of Application Traffic
- Quality of Service Renegotiation

Initial QoS

When the session is established, an initial level of QoS must be assigned to the subscriber. The GGSN may either grant the requested QoS, or grant a lower QoS level (minimum or intermediate level). The initial QoS remains in effect until the SGSN or GGSN requests a change. When Dynamic QoS Renegotiation is enabled, there are several conditions when the system would request a QoS change.

- Services detected that do not need high QoS: After a configurable time period of a subscriber having terminated services that require high QoS, the system could lower the QoS to a value more appropriate to the services actually being used.
- Services detected that require higher QoS: As soon as a subscriber begins using a service that needs a high QoS, the system immediately attempts to raise the QoS through its service detection capability.

Service Detection

The Application analysis approach to service detection uses application level (L7) information. In the ASR 5000 chassis, application analysis is stateful—keeping track of the application state.



Important: For L7 traffic analysis ECSv2 license is required.

Classification of Application Traffic

Application traffic can be classified into the following: Conversational, Streaming, Interactive 1, Interactive 2, Interactive 3, or Background. Traffic class can be configured in the charging-action, but it does not take direction as a parameter. However, you can configure a rule matching uplink-only or downlink-only packets and associate it with the charging-action.

QoS renegotiation requires knowing what kind of data packets are flowing through for a particular user to associate a given traffic class with the user's current usage pattern. This is done through packet inspection for a subscriber profile via an Access Control List (ACL). Limits for each traffic class can be configured in the APN. The same infrastructure is reused to perform Dynamic QoS Renegotiation.

After classification of traffic and if required by subscriber profile, Dynamic QoS Renegotiation takes place.

L4 Packet Inspection

L4 packet analysis has no or low impact on the system performance with very limited impact on system capacity. L4 packet inspection is fully supported by the system.

L7 Packet Inspection

L7 packet analysis has a greater impact on system performance with very limited impact on the system capacity. L7 packet inspection involves complete application layer analysis and copes with customized applications.

QoS Renegotiation for a Subscriber QoS Profile

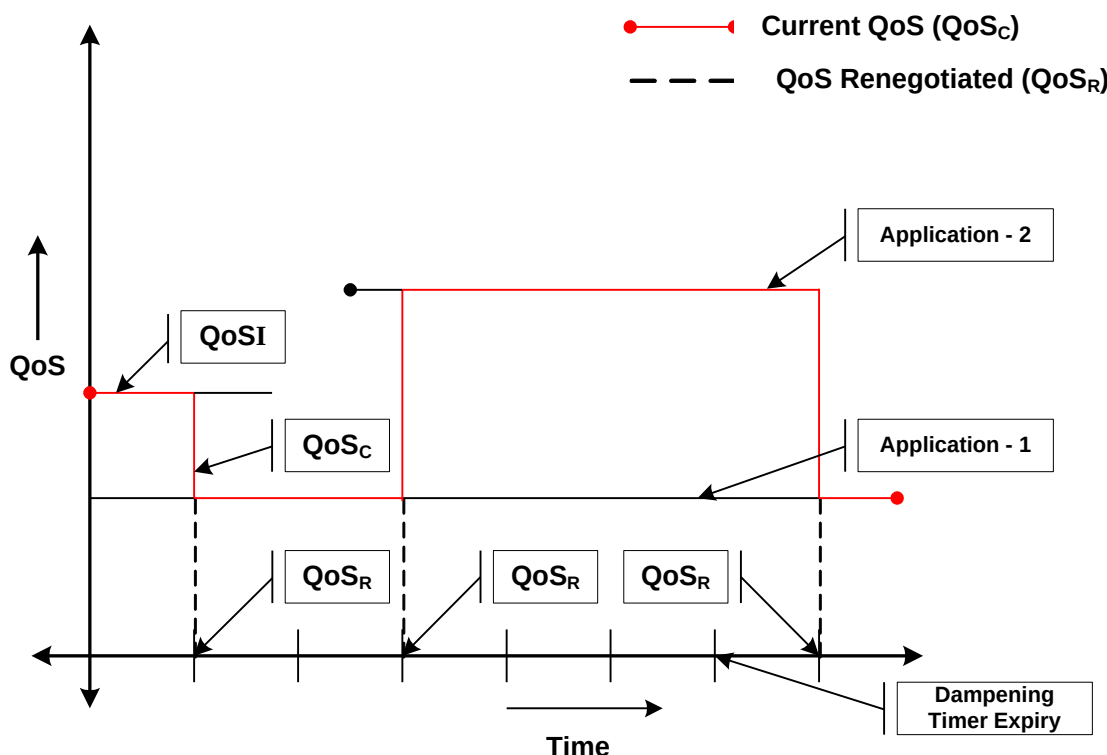
The following is the overall Dynamic QoS Renegotiation process.

1. When a subscriber attaches to the network, the following happens:
 - Dampening timer is started for the subscriber.
 - QoSI is assigned to the subscriber. This becomes the QoSC till a re-negotiation occurs, as shown in the figure below.
 - The traffic class bitfield is cleared.
2. As the subscriber starts using some applications, the traffic gets classified on the basis of type of data packets or traffic as mentioned in section *Classification of Application Traffic*. The corresponding bit in the Traffic-class-bitfield is set accordingly.
3. The mechanics of QoS renegotiation are as follows:

- Examine traffic-class-bitfield to determine the highest bit that is set. This gives the desired QoS Traffic Class (QoSD). The associated uplink/downlink peak-data-rate and guaranteed-data-rate values are taken from the configured parameters for this traffic class in the subscriber APN.
 - If QoSC matches QoSD, no QoS renegotiation is required. Otherwise, send an Update PDP Context Request to the SGSN with the QoSD values and QoS renegotiation starts.
 - Reset the dampening timer.
 - Clear the traffic-class-bitfield.
4. QoS renegotiation happens under the following conditions:
- When a higher priority traffic is detected, QoS is renegotiated immediately without waiting for the dampening time to expire. For example, if the current traffic has a QoS of Interactive and the system detects streaming traffic, QoS is immediately upgraded to Streaming.
 - When lower priority traffic is detected, the system waits for the expiry of the dampening timer before lowering the QoS.
 - During “silence” or no-traffic, QoS renegotiation requests are not initiated.

As seen in the following figure, the QoS profile for the subscriber goes through three renegotiations to match the QoS profile of the highest priority application currently being used.

5. Dynamic QoS Renegotiation Graph



When there is no traffic, traffic class drops to “Background” and the corresponding QoS profile is negotiated as described above.

Network Controlled QoS (NCQoS)

Network-controlled QoS is the method by which the system updates the QoS for a PDP context (primary or secondary) upon receipt of Network Requested Update PDP Context (NRUPC) messages from the GGSN. The system can also activate a new secondary PDP context upon receipt of a Network Requested Secondary PDP Context Activation (NRSPCA) message from the GGSN.

How Network Controlled QoS (NCQoS) Works

The GGSN activates or modifies a bearer whenever a service flow matches a statically provisioned Policy and Charging Control (PCC) rule. The network, based on QoS requirements of the application/service, determines what bearers are needed and either modifies an existing bearer or activates a new one.

Statically provisioned PCC rules, called Network Requested Operation (NRO) rules, are configured as charging rules in the Active Charging Service (ACS). As a part of charging action for such rules, QoS-needed and corresponding Traffic Flow Template (TFT) packet filters are configured. QoS-needed mainly consists of QoS Class Identifier (QCI) and data rates. Whereas, TFT mainly consists of uplink and downlink packet filter information.



WARNING: This feature does not work in conjunction with IMS-Authorization service.

When a packet arrives, the ACS analyzes it and performs rule matching based on the priority in the rulebase. If an NRO rule bound to the context on which the packet arrived matches, ACS applies the bandwidth limit and gating. If an NRO rule bound to some other context matches, ACS discards the packet.

If an unbound NRO rule matches, ACS finds a context with the same QCI as the NRO rule, where the context's Maximum Bit Rate (MBR) and matched rule's MBR (context's MBR + matched rule's MBR) is less than the MBR for that QCI in the APN. If such a context is found, NRUPC for that context is triggered. If the request succeeds, the rule will be bound to that context.



Important: The packet that triggered the NRUPC request is discarded.

If no context satisfying the MBR limit is found, or if there is no context with the same QCI as the NRO rule, the system triggers NRSPCA. If the request succeeds, the rule is bound to that context.



Important: The packet that triggered the NRSPCA request is discarded.

TFTs from the charging-action associated with the NRO rule are also sent as part of the NRUPC/NRSPCA request, and returned as part of the Create PDP Context Response.

Finally, if a non-NRO rule matches, ACS proceeds with the normal processing of that packet. Non-NRO charging-actions can still do “flow action” or ITC (limit-for-flow-type and limit-for-bandwidth).

ACS also does the following:

- Before making an NRUPC/NRSPCA Request, ACS checks if there is any outstanding request for the same QCI for the same subscriber. If there is, it will not process the new request and discards the packet.
- After a context is terminated, ACS unbinds all the rules bound to that context. Such a rule can later be bound to some other context when a packet matches that rule.



Important: The packet that triggered the NRUPC/NRSPCA request is discarded.

Configuring Dynamic QoS Renegotiation

This section describes how to configure per-APN based Dynamic QoS Renegotiation.



Caution: For Dynamic QoS Renegotiation, two RADIUS attributes are required for remote subscriber configuration. For a particular subscriber, these attributes can be overridden without considering the timeout for Dynamic QoS Renegotiation and whether Dynamic QoS Renegotiation is enabled or not.

To configure Dynamic QoS Renegotiation:

- Step 1** Configure an Access Control List (ACL), as described in the [Configuring ACL for Dynamic QoS Renegotiation](#) section.
- Step 2** Configure an APN for Dynamic QoS Renegotiation as described in the [Configuring APNs for Dynamic QoS Renegotiation](#) section.
- Step 3** Save your configuration as described in the *Verifying and Saving Your Configuration* chapter in this guide.
- Step 4** Monitor the operations as described in the [Monitoring Dynamic QoS Renegotiation Operation](#) section.



Important: Commands used in the configuration examples in this section reflect base functionality (most common or likely commands and/or keyword options). In many cases, other commands and/or keyword options are available. Refer to the *ACS Configuration Mode Commands* and *APN Configuration Mode Commands* sections of the *Command Line Interface Reference* for complete information regarding all commands.

Configuring ACL for Dynamic QoS Renegotiation

Configuring an ACL and applying it to an APN template are required to specify permission and treatment levels for Dynamic QoS Renegotiation.

Use the following example to configure an ACL for Dynamic QoS Renegotiation:

```
configure
  context <context_name>
    ip access-list <acl_name>
      permit { tcp | udp } ..... treatment { background | conversational |
interactive-1 | interactive-2 | interactive-3 | streaming }
    end
```

Notes:

- *context_name* must be the name of the destination context in which you want to configure the ACL. The same context must be used for APN configuration.

- For information on configuring the rules that comprise the ACL, refer to the *Access Control Lists* appendix.

Configuring Charging Action for Dynamic QoS Renegotiation

Use the following example to configure charging action parameters for Dynamic QoS Renegotiation support:

```
configure

  active-charging service <service_name>

    charging-action <charging_action_name> -noconfirm

    qos-renegotiate traffic-class streaming

    flow action discard

    flow limit-for-bandwidth direction downlink peak-data-rate <bps> peak-
burst-size <bytes> violate-action lower-ip-precedence

  end
```

Notes:

- A maximum of eight packet filters can be configured per charging action.
- The flow limit-for-bandwidth command contains other option than the example shown here. Refer ti the *ACS Charging Action Configuration Mode Commands* chapter in the *Command Line Interface Reference* for more information on this command.

Configuring Rulebase for Dynamic QoS Renegotiation

Use the following example to configure rulebase parameters for Dynamic QoS Renegotiation support:

```
configure

  active-charging service <service_name>

    rulebase <rulebase_name> [ -noconfirm ]

    qos-renegotiate timeout <timeout>

  end
```

Configuring APNs for Dynamic QoS Renegotiation

Use the following example to configure an APN template's QoS profile in support of Dynamic QoS Renegotiation:

```
configure

  context <context_name>

    apn <apn_name>
```

```
ip access-group <acl_name> [ in | out ]  
  
end
```

Notes:

- *context_name* must be the name of the destination context in which you have already configured the ACL, and want to configure the APN template.
- *<acl_name>* must be the name of the ACL that you have already configured in the context.
- If the optional **in** or **out** keywords are not specified in the **ip access-group** command (APN Configuration Mode), the ACL will be applied to all inbound and outbound packets.

Configuring Network Controlled QoS (NCQoS)

To configure NCQoS:

- Step 1** Configure packet filter parameters as described in the [Configuring Packet Filter for NCQoS](#) section.
- Step 2** Configure charging rules and actions as described in the [Configuring Charging Action for NCQoS](#) section.
- Step 3** Configure APN template and enable bearer control mode for NCQoS as described in the [Configuring APN for NCQoS](#) section.
- Step 4** Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.
- Step 5** Monitor the operations as described in the [Monitoring Dynamic QoS Renegotiation Operation](#) section.



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

Configuring Packet Filter for NCQoS

Use the following example to configure packet filter parameters for NCQoS support:

configure

```

active-charging service <service_name>

  packet-filter <filter_name> [ -noconfirm ]

    ip local-port { = <port_num> | range <start_port_num> to <end_port_num> }

    ip protocol { = <proto_num> | range <start_proto_num> to <end_proto_num> }

    ip remote-address { = { <ip_address> | <ip_address/mask> } | { range {
<ip_address> | <ip_address/mask> } to { <ip_address> | <ip_address/mask> } }

    ip remote-port { = <port_num> | range <start_port_num> to <end_port_num> }

    direction { bi-directional | download | upload }

    priority <priority>

  end

```

Configuring Charging Action for NCQoS

Use the following example to configure charging action parameters for NCQoS support:

```
configure

  active-charging service <service_name>

    charging-action <charging_action_name> [ -noconfirm ]

    qos-class-identifier <identifier>

    flow action discard [ downlink | uplink ]

    tft packet-filter <filter_name>

    flow limit-for-bandwidth direction { downlink | uplink } peak-data-rate <bps>
peak-burst-size <bytes> violate-action { discard | lower-ip-precedence }

  end
```

Notes:

- A number of optional keywords and variable are available for the **flow limit-for-bandwidth direction** command. Refer to the *ACS Charging Action Configuration Mode Commands* section of the *Command Line Interface Reference* for more information regarding this command.

Configuring APN for NCQoS

Use the following example to enable Bearer Control Mode (BCM) for NCQoS support:

```
configure

  context <context_name>

    apn <apn_name>

    bearer-control-mode [ mixed | ms-only | none ]

  end
```

Notes:

- To enable NCQoS, bearer-control-mode in the APN Configuration Mode must be configured with **mixed** mode.

Monitoring Dynamic QoS Renegotiation Operation

Use the following steps to verify/monitor Dynamic QoS Renegotiation operations:

Step 1 Verify that your APNs were configured properly by entering the following command:

```
show apn { all | name apn_name }
```

The output is a listing of APN parameter settings.

Step 2 Verify that the ACLs have been properly applied by entering the following command:

```
show apn name apn_name
```

apn_name must be the name of the APN configured in the *Configuring APNs for Dynamic QoS Renegotiation* section. The output of this command displays the APN configuration. Examine the output for the **ip output access-group** and **ip input access-group** fields. For more details refer to the *Applying a Single ACL to Multiple Subscribers* section.

Step 3 Verify that your ACL was configured properly by entering the following command:

```
show ip access-list acl_name
```

The output is a concise listing of IP Access Control List parameter settings.

Step 4 Monitor your QoS renegotiation status for a subscriber by running the **show subscriber ggsn-only full** command (Exec mode).

The output is a concise listing of subscribers' settings.

Step 5 For L7 based QoS Renegotiation, view how many time QoS renegotiations have happened for that session by running the **show active-charging sessions full all** command (Exec mode).

Step 6 View the statistics of APN related to QoS renegotiation parameters by entering the following command:

```
show apn statistics { all | name apn_name }
```

The output is a listing of APN statistics related to QoS Renegotiation.

Event IDs Pertaining to Dynamic QoS Renegotiation

The Session Manager facility sources event IDs that can be useful for diagnosing errors that could occur when implementing of Dynamic QoS Renegotiation feature.

The following table displays information pertaining to these events.

Table 53. Event IDs in Session Manager Pertaining to Dynamic QoS Renegotiation

Event	Event ID	Type	Additional Information
QoS Renegotiation timer started for subscriber	10917	Info	“Indicates that the Dynamic QoS Renegotiation timer was started for the subscriber”
QoS Renegotiation timer stopped for subscriber	10918	Info	“Indicates that the Dynamic QoS Renegotiation timer was stopped for the subscriber”
QoS Renegotiation timer expired for subscriber	10919	Info	“Indicates that the Dynamic QoS Renegotiation timer was expired for the subscriber”
QoS Renegotiation message sent for subscriber	10920	Info	“Indicates that the Dynamic QoS Renegotiation message was sent for the subscriber”
L4 classification done for subscriber traffic	10921	Info	“Indicates the kind of L4 classification that was done for the subscriber traffic.”

RADIUS Attributes

The RADIUS attributes listed in the following table are used to enable Dynamic QoS Renegotiation for subscribers configured on remote RADIUS servers. More information on these attributes can be found in the *AAA and GTPP Interface Administration and Reference*.

Table 54. RADIUS Attributes Required for Dynamic QoS Renegotiation Support

Attribute	Description
SN-Enable-QoS-Renegotiation (or SN1-Enable-QoS-Renegotiation)	Enables the Dynamic QoS Renegotiation for specific profile application. This attribute displays “enable qos renegotiation”.
SN-QoS-Renegotiation-Timeout (or SN1-QoS-Renegotiation-Timeout)	Timeout duration for dampening time for QoS renegotiation to specific profile application. This attribute displays “qos renegotiation timeout”.

Appendix H

Routing

This appendix provides information on configuring an enhanced, or extended, service. The product administration guides provide examples and procedures for configuring basic services on the system. You should select the configuration example that best meets your service model, and configure the required elements for that model before using the procedures in this appendix.

This appendix includes the following sections:

- [Routing Policies](#)
- [Static Routing](#)
- [OSPF Routing](#)
- [OSPFv3 Routing](#)
- [Viewing Routing Information](#)
- [Equal Cost Multiple Path \(ECMP\)](#)
- [BGP-4 Routing](#)
- [Bidirectional Forwarding Detection](#)

Routing Policies

This section describes how to configure the elements needed to define routing policies. Routing policies modify and redirect routes to and from the system to satisfy specific network deployment requirements.

Use the following building blocks to configure routing policies:

- **Route Access Lists** – The basic building block of a routing policy. Route access lists filter routes based on a range of IP addresses.
- **IP Prefix Lists** – A more advanced element of a routing policy. An IP Prefix list filters routes based on IP prefixes.
- **AS Path Access Lists** – A basic building block used for Border Gateway Protocol (BGP) routing. These lists filter Autonomous System (AS) paths.
- **Route Maps** – Route-maps provide detailed control over routes during route selection or route advertisement by a routing protocol, and in route redistribution between routing protocols. For this level of control you use IP Prefix Lists, Route Access Lists and AS Path Access Lists to specify IP addresses, address ranges, and Autonomous System paths.

Creating IP Prefix Lists

Use the following configuration example to create IP Prefix Lists:

```
config
context <context_name>
    ip prefix-list name <list_name> { deny | permit } <network_address/net_mask>
```

Notes:

- Set the IP prefix list to deny, permit or match any prefix.
- IPv4 dotted-decimal and IPv6 colon-separated addresses are supported.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

Creating Route Access Lists

Use the following procedure to create a Route Access List:

```
config
context <context_name>
    route-access-list { extended identifier } { deny | permit } [ ip address
<ip_address> ]
    route-access-list named <list_name> { deny | permit } { <ip_address/mask> | any }
[ exact-match ]
```

```
route-access-list standard <identifier> { permit | deny } { <ip_address> <wildcard_mask>
| any | <network_address> }
```

Notes:

- A maximum of 64 access lists are supported per context.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

Creating AS Path Access Lists

Use the following procedure to create an AS Path Access List:

```
config

context <context_name>

ip as-path access-list <list_name> [ { deny | permit } <reg_expr> ]
```

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

Creating Route Maps

Use the following configuration example to create a Route Map:

```
config

context <context_name>

route-map <map_name> { deny | permit } <seq_number>
```

Notes:

- Use the **match** and **set** commands in Route Map Configuration mode to configure the route map. Refer to the *Command Line Interface Reference* for more information on these commands.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

Sample Configuration

The example below shows a configuration that creates two route access lists, applies them to a route map, and uses that route map for a BGP router neighbor.

```
config

context isp1

route-access-list named RACLin1a permit 88.151.1.0/30

route-access-list named RACLin1a permit 88.151.1.4/30

route-access-list named RACLany permit any
```

```
route-map RMnet1 deny 100

  match ip address route-access-list RACLin 1 a

  #exit

route-map RMnet1 deny 200

  match ip address route-access-list RACLin 1 b

  #exit

route-map RMnet1 permit 1000

  match ip address route-access-list RACLany

  #exit

router bgp 1

  neighbor 152.20.1.99 as-path 101

  neighbor 152.20.1.99 route-map RMnet1
```

Static Routing

The system supports static network route configuration on a per context basis. Define network routes by specifying the:

- IP address and mask for the route
- Name of the interface in the current context that the route must use
- Next hop IP address

Adding Static Routes to a Context

To add static routes to a context configuration, you must know the names of the interfaces that are configured in the current context. Use the **show ip interface** command to list the interfaces in the current context (Exec mode).

Information for all interfaces configured in the current context is displayed as shown in the following example.

```
[local]<host_name># show ip interface

Intf Name: Egress 1

Description:

IP State: Up (Bound to 24/1 untagged ifIndex 402718721)

IP Address: 192.168.231.5

Subnet Mask: 255.255.255.0

Bcast Address: 192.168.231.255

MTU: 1500

Resoln Type: ARP ARP timeout: 3600 secs

L3 monitor LC-port switchover: Disabled

Number of Secondary Addresses: 0

Total interface count: 1
```

The first line of information for each interface lists the interface name for the current context as shown in the example output. In this example, there is one interface with the name *Egress 1*.

config

```
context <context_name>

    ip route { <ip_address> [ <ip_mask> ] | <ip_addr_mask_combo> } { next-hop
<next_hop_address> | <egress_name> [ precedence <precedence> [ cost <cost> ]
```

Notes:

You can configure a maximum of 1,200 static routes per context. Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

Deleting Static Routes From a Context

Use the following configuration example to remove static routes from a context's configuration:

config

```
context context_name

    no ip route { <ip_address> <ip_mask> | <ip_addr_mask_combo> } <next_hop_address>
    <egress_name> [ precedence <precedence> ] [ cost <cost> ]
```

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

OSPF Routing

This section gives an overview of Open Shortest Path First (OSPF) routing and its implementation in the system. It also describes how to enable the base OSPF functionality and lists the commands that are available for more complex configurations.

You must purchase and install a license key before you can use this feature. Contact your Cisco account representative for more information on licenses.



Important: During system task recovery, it is possible for a dynamically-learned forwarding entry to incorrectly remain in the system forwarding table if that forwarding entry has been removed from the dynamic routing protocol during the recovery.

OSPF Version 2 Overview

OSPF is a link-state routing protocol that employs an interior gateway protocol (IGP) to route IP packets using the shortest path first based solely on the destination IP address in the IP packet header. OSPF routed IP packets are not encapsulated in any additional protocol headers as they transit the network.

An Autonomous System (AS), or Domain, is defined as a group of networks within a common routing infrastructure.

OSPF is a dynamic routing protocol that quickly detects topological changes in the AS (such as router interface failures) and calculates new loop-free routes after a period of convergence. This period of convergence is short and involves a minimum of routing traffic.

In a link-state routing protocol, each router maintains a database, referred to as the link-state database, that describes the Autonomous System's topology. Each participating router has an identical database. Each entry in this database is a particular router's local state (for example, the router's usable interfaces and reachable neighbors). The router distributes its local state throughout the AS by flooding.

All routers run the same algorithm in parallel. From the link-state database, each router constructs a tree of shortest paths with itself as root to each destination in the AS. Externally derived routing information appears on the tree as leaves. The cost of a route is described by a single dimensionless metric.

OSPF allows sets of networks to be grouped together. Such a grouping is called an area. The topology of this area is hidden from the rest of the AS, which enables a significant reduction in routing traffic. Also, routing within the area is determined only by the area's own topology, lending the area protection from bad routing data. An area is a generalization of an IP subnetted network.

OSPF enables the flexible configuration of IP subnets so that each route distributed by OSPF has a destination and mask. Two different subnets of the same IP network number may have different sizes (that is, different masks). This is commonly referred to as variable-length subnetting. A packet is routed to the best (longest or most specific) match. Host routes are considered to be subnets whose masks are "all ones" (0xffffffff).

OSPF traffic can be authenticated or non-authenticated, or can use no authentication, simple/clear text passwords, or MD5-based passwords. This means that only trusted routers can participate in the AS routing. You can specify a variety of authentication schemes and, in fact, you can configure separate authentication schemes for each IP subnet.

Externally derived routing data (for example, routes learned from an exterior protocol such as BGP) is advertised throughout the AS. This externally derived data is kept separate from the OSPF link state data.

Each external route can also be tagged by the advertising router, enabling the passing of additional information between routers on the boundary of the AS.

OSPF uses a link-state algorithm to build and calculate the shortest path to all known destinations.

Basic OSPFv2 Configuration

This section describes how to implement basic OSPF routing.

Enabling OSPF Routing For a Specific Context

Use the following configuration example to enable OSPF Routing for a specific context:

```
config
    context <context_name>
        router ospf
    end
```

Notes:

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

Enabling OSPF Over a Specific Interface

After you enable OSPF, specify the networks on which it will run. Use the following command to enable OSPF:

```
network network_ip_address/network_mask area { area_id | area_ip_address }
```



Important: The default cost for OSPF on the system is 10. To change the cost, refer to the **ip ospf cost** command in the *Ethernet Interface Configuration Commands* section of the *Command Line Interface Reference*.

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

Redistributing Routes Into OSPF (Optional)

Redistributing routes into OSPF means any routes from another protocol that meet specified a specified criterion, such as route type, metric, or rule within a route-map, are redistributed using the OSPFv2 protocol to all OSPF areas. This is an optional configuration.

```
config
    context <context_name>
        router ospf
            redistribute { connected | static }
        end
```


Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

Confirming OSPF Configuration Parameters

To confirm the OSPF router configuration, use the following command and look for the section labeled **router ospf** in the screen output:

```
show config context ctxt_name [ verbose ]
```

OSPFv3 Routing

This section gives an overview of Open Shortest Path First Version 3 (OSPFv3) routing and its implementation in the system. It also describes how to enable the base OSPFv3 functionality and lists the commands that are available for more complex configurations.

OSPFv3 Overview

Much of OSPF version 3 is the same as OSPF version 2. OSPFv3 expands on OSPF version 2 to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses. OSPFv3 dynamically learns and advertises (redistributes) IPv6 routes within an OSPFv3 routing domain.

In OSPFv3, a routing process does not need to be explicitly created. Enabling OSPFv3 on an interface will cause a routing process and its associated configuration to be created.

Basic OSPFv3 Configuration

This section describes how to implement basic OSPF routing.

Enabling OSPFv3 Routing For a Specific Context

Use the following configuration example to enable OSPF Routing for a specific context:

```
config
    context <context_name>
        router ospfv3
    end
```

Notes:

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

Enabling OSPFv6 Over a Specific Interface

After you enable OSPFv3 specify the area in which it will run. Use the following command to enable OSPFv3:

```
area { area_id | area_ip_address } [ default-cost dflt-cost ] [ stub stub-area ] [
virtual-link vl-neighbor-ipv4address ]
```



Important: The default cost for OSPFv3 on the system is 10. To change the cost, refer to the **ipv6 ospf cost** command in the *Ethernet Interface Configuration Commands* section of the *Command Line Interface Reference*.

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

Redistributing Routes Into OSPFv3 (Optional)

Redistributing routes into OSPFv3 means any routes from another protocol that meet specified a specified criterion, such as route type, metric, or rule within a route-map, are redistributed using the OSPFv3 protocol to all OSPF areas. This is an optional configuration.

```
config
    context <context_name>
        router ospf3
            redistribute { connected | static }
        end
```

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

Confirming OSPFv3 Configuration Parameters

To confirm the OSPF router configuration, use the following command and look for the section labeled **router ipv6 ospf** in the screen output:

```
show config context <ctxt_name> [ verbose ]
```

Viewing Routing Information

To view routing information for the current context, run one of the following Exec mode commands;

- **show ip route**: Displays information for all types of routes in the current contexts routing table.
- **show ip static-route**: Displays information only for static routes in the current contexts routing table.
- **show ip ospf**: Displays OSPF process summary information in the current context.
- **show ipv6 ospf**: Displays OSPFv3 process summary information in the current context.

This example shows sample output of the command, **show ip route**.

```
[local]host_name# show ip route

"*" indicates the Best or Used route. Destination Nexthop Protocol Prec Cost
Interface

*44.44.44.0/24 208.230.231.50 static 1 0 local1

*192.168.82.0/24 0.0.0.0 connected 0 0

*192.168.83.0/24 0.0.0.0 connected 0 0

208.230.231.0/24 0.0.0.0 ospf 110 10 local1

*208.230.231.0/24 0.0.0.0 connected 0 0 local1

Total route count: 5
```

Equal Cost Multiple Path (ECMP)

The system supports ECMP for routing protocols. ECMP distributes traffic across multiple routes that have the same cost to lessen the burden on any one route.

config

```
context <context_name>  
  
    ip routing maximum-paths [ max_num ]
```

Notes:

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

BGP-4 Routing

The Border Gateway Protocol 4 (BGP-4) routing protocol is supported through a BGP router process that is implemented at the context level.

The Border Gateway Protocol (BGP) is an inter-AS routing protocol. An Autonomous System (AS) is a set of routers under a single technical administration that use an interior gateway protocol and common metrics to route packets within the AS. The set of routers uses an exterior gateway protocol to route packets to other autonomous systems.

BGP runs over TCP. This eliminates the need for the BGP protocol to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing information. Any authentication scheme used by TCP may be used in addition to BGP's own authentication mechanisms.

BGP routers exchange network reachability information with other BGP routers. This information builds a picture of AS connectivity from which routes are filtered and AS-level policy decisions are enforced.

BGP-4 provides classless inter-domain routing. This includes support for advertising an IP prefix and eliminates the concept of network class within BGP. BGP-4 also allows the aggregation of routes, including the aggregation of AS paths.

Overview of BGP Support

Mobile devices communicate to the Internet through Home Agents (HAs). HAs assign IP addresses to the mobile node from a configured pool of addresses. These addresses are also advertised to Internet routers through an IP routing protocol to ensure dynamic routing. The BGP-4 protocol is used as a monitoring mechanism between an HA and Internet router with routing to support Interchassis Session Recovery (ICSR). (Refer to the *Interchassis Session Recovery* appendix in this guide for more information.)

The objective of BGP-4 protocol support is to satisfy routing requirements and monitor communications with Internet routers. BGP-4 may trigger an active to standby switchover to keep subscriber services from being interrupted.

The following BGP-4 features are supported:

- Exterior Border Gateway Protocol (EBGP) multi-hop
- Route Filtering for inbound and outbound routes
- Route redistribution and route-maps

IP pool routes and loopback routes are advertised in the BGP domain in the following ways:

- Through BGP Configuration Mode **redistribution** commands, all or some of the connected routes are redistributed into the BGP domain. (IP pool and loopback routes are present in the IP routing table as connected routes.) The **network routemap** command provides the flexibility to change many BGP attributes.
- Through the BGP Configuration Mode **network** commands, connected routes are explicitly configured for advertisement into the BGP domain. The **network routemap** command provides the flexibility to change many BGP attributes. Refer to the *Border Gateway Protocol Configuration Mode Commands* section of the *Command Line Interface Reference* for details on these commands.



Important: If a BGP task restarts because of a processing card failure, a migration, a crash, or the removal of a processing card, all peering session and route information is lost.

Configuring BGP

This section describes how to configure and enable basic BGP routing support in the system.

config

```
context <context_name>

  router bgp <AS_number>

    neighbor <ip_address> remote-as <AS_num>
```

Notes:

- A maximum of 64 BGP peers are supported per context.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

Redistributing Routes Into BGP (Optional)

Redistributing routes into BGP simply means that any routes from another protocol that meet a specified criterion, such as a route type, or a rule within a route-map, are redistributed through the BGP protocol to all BGP areas. This is an optional configuration.

config

```
context <context_name>

  router bgp <as_number>

    redistribute bgp { bgp | connected | static } [ metric <metric_value> ]
  [ metric-type { 1 | 2 } ] [ route-map <route_map_name> ]
```

Notes:

- The redistribution options are connected, ospf, rip, or static. Refer to the *Border Gateway Protocol Configuration Mode Commands* section of the *Command Line Interface Reference* for details on the **redistribute** command.
- A maximum of 64 route-maps are supported per context.
- Save your configuration as described in *Verifying and Saving Your Configuration* chapter of this guide.

Bidirectional Forwarding Detection

Bidirectional Forwarding Detection (BFD) is a network protocol used to detect faults between two forwarding engines connected by a link. BFD establishes a session between two endpoints over a particular link. If more than one link exists between two systems, multiple BFD sessions may be established to monitor each one of them. The session is established with a three-way handshake, and is torn down the same way. Authentication may be enabled on the session. A choice of simple password, MD5 or SHA1 authentication is available.

Overview of BFD Support

BFD does not have a discovery mechanism; sessions must be explicitly configured between endpoints. BFD may be used on many different underlying transport mechanisms and layers, and operates independently of all of these. Therefore, it needs to be encapsulated by whatever transport it uses.

In asynchronous mode, both endpoints periodically send Hello packets to each other. If a number of those packets are not received, the session is considered down.

When Echo is active, a stream of Echo packets is sent to the other endpoint which then forwards these back to the sender. Echo can be globally enabled via the **bfd-protocol** command, and/or individually enabled/disabled per interface. This function is used to test the forwarding path on the remote system.

The ASR 5000 supports BFD in asynchronous mode with optional Echo capability via static or BGP routing.

Configuring BFD

This section describes how to configure and enable basic BFD routing protocol support in the system.

There are several factors affecting the configuration of BFD protocol:

- [Configuring a BFD Context](#)
- [Configuring IPv4 BFD for Static Routes](#)
- [Configuring BFD for BGP Single Hop](#)
- [Configuring Multihop BFD for BGP](#)
- [Scaling of BFD](#)

Configuring a BFD Context

```
config
    context <context_name>
        bfd-protocol
            [ [no] bfd echo ]
    exit
```


Notes:

- Echo function can be optionally enabled for all interfaces in this context.
- 16 BFD sessions per context and 64 per chassis.

Configuring IPv4 BFD for Static Routes

Enable BFD on an interface.

config

```
context <bfd_context_name>

    interface <if_name>

        ip address <ipv4_address> <ipv4_mask>

        bfd interval <interval_value> min_rx <rx_value> multiplier <multiplier_value>

        [ [no] bfd echo ]

    exit
```

Configure BFD static route.

```
ip route static bfd <if_name> <ipv4_gw_address>
```

Add static routes.

```
ip route <ipv4_address> <ipv4_mask>

ip route <ipv4_address> <ipv4_mask>
```

Configuring BFD for BGP Single Hop

Enable BFD on an interface.

config

```
context <bfd_context_name>

    interface <if_name>

        ip address <ipv4_address> <ipv4_mask>

        bfd interval <interval_value> min_rx <rx_value> multiplier <multiplier_value>

        [ [no] bfd echo ]

    exit
```

Enable BFD on a BGP Neighbor. See [Associate BGP Neighbors with the Context](#) for additional information.

```
router bgp <AS_number>
```

```
neighbor <neighbor_ip-address> failover bfd
```

Configuring Multihop BFD for BGP

Enable BFD on an interface.

```
config
```

```
context <bfd_context_name>
```

```
interface <if_name>
```

```
ip address <ipv4_address> <ipv4_mask>
```

```
bfd interval <interval_value> min_rx <rx_value> multiplier <multiplier_value>
```

```
[ [no] bfd echo ]
```

```
exit
```

Configure a Multihop BFD session.

```
bfd-protocol
```

```
bfd multihop peer <destination-address> interval <interval-value> multiplier
<multiplier-value>
```

Enable BFD on a BGP Neighbor. See [Associate BGP Neighbors with the Context](#) for additional information.

```
neighbor <neighbor_ip-address> failover bfd multihop
```

Scaling of BFD

Configure an active BFD session using one of the above methods and use same BFD neighbor while configuring the active interface. See [Associate BFD Neighbor Groups with the BFD Protocol](#) for additional information.

```
bfd-protocol
```

```
bfd nbr-group-name <grp_name> active-if-name <if_name> <nexthop_address>
```

Apply the same BFD results to one or more passive interfaces.

```
bfd nbr-group-name <grp_name> passive-if-name <if_name> <nexthop_address>
```

```
bfd nbr-group-name <grp_name> passive-if-name <if_name> <nexthop_address>
```

Associate BGP Neighbors with the Context

```
config

context <context_name>

    router bgp <AS_number>

        neighbor <neighbor_ip-address> remote-as <rem_AS_number>

        neighbor <neighbor_ip-address> ebgp-multihop max-hop <max_hops>

        neighbor <neighbor_ip-address> update-source <update-src_ip-address>

        neighbor <neighbor_ip-address> failover bfd [ multihop ]
```

Notes:

- Repeat the sequence to add neighbors.

Associate BFD Neighbor Groups with the BFD Protocol

```
config

context <context_name>

    bfd-protocol

        bfd nbr-group-name <grp_name> active-if-name <if_name> <nexthop_address>

        bfd nbr-group-name <grp_name> passive-if-name <if_name> <nexthop_address>
```



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

Appendix I

Session Recovery

With robust hardware failover and redundancy protection, any card-level hardware failures on the system can quickly be corrected. However, software failures can occur for numerous reasons, often without prior indication.

This appendix describes the Session Recovery feature that provides seamless failover and reconstruction of subscriber session information in the event of a hardware or software fault.

This appendix includes the following sections:

- [How Session Recovery Works](#)
- [Additional Hardware Requirements](#)
- [Configuring the System to Support Session Recovery](#)

How Session Recovery Works

This section provides an overview of how this feature is implemented and the recovery process.

The Session Recovery feature provides seamless failover and reconstruction of subscriber session information in the event of a hardware or software fault within the system preventing a fully connected user session from being disconnected.

Session recovery is performed by mirroring key software processes (for example, session manager and AAA manager) within the system. These mirrored processes remain in an idle state (standby-mode) wherein they perform no processing, until they may be needed in the event of a software failure (for example, a session manager task aborts).

The system spawns new instances of “standby mode” session and AAA managers for each active control processor (CP) being used. These mirrored processes require both memory and processing resources, which means that additional hardware may be required to enable this feature (see the [Additional Hardware Requirements](#) section).

Other key system-level software tasks, such as VPN manager, are performed on a physically separate packet processing card to ensure that a double software fault (for example, session manager and VPN manager fails at same time on same card) cannot occur. The packet processing card that hosts the VPN manager process is in active mode and reserved by the operating system for this sole use when session recovery is enabled.

There are two modes of session recovery.

- **Task recovery mode:** Wherein one or more session manager failures occur and are recovered without the need to use resources on a standby packet processing card. In this mode, recovery is performed by using the mirrored “standby-mode” session manager task(s) running on active packet processing cards. The “standby-mode” task is renamed, made active, and is then populated using information from other tasks such as AAA manager. In case of Task failure, limited subscribers will be affected and will suffer outage only until the task starts back up.
- **Full packet processing card recovery mode:** Used when a packet processing card hardware failure occurs, or when a planned packet processing card migration fails. In this mode, the standby packet processing card is made active and the “standby-mode” session manager and AAA manager tasks on the newly activated packet processing card perform session recovery.

Session/Call state information is saved in the peer AAA manager task because each AAA manager and session manager task is paired together. These pairs are started on physically different packet processing cards to ensure task recovery.

There are some situations wherein session recovery may not operate properly. These include:

- Additional software or hardware failures occur during the session recovery operation. For example, an AAA manager fails while the state information it contained was being used to populate the newly activated session manager task.
- A lack of hardware resources (packet processing card memory and control processors) to support session recovery.



Important: After a session recovery operation, some statistics, such as those collected and maintained on a per manager basis (AAA Manager, Session Manager, etc.) are in general not recovered, only accounting and billing related information is checkpointed and recovered.

Session Recovery is available for the following functions:

- Any session needing L2TP LAC support (excluding regenerated PPP on top of an HA/GGSN session)
- Closed RP PDSN services supporting simple IP, Mobile IP, and Proxy Mobile IP

- CSCF sessions
- GGSN services for IPv4 and PPP PDP contexts
- HA services supporting Mobile IP and/or Proxy Mobile IP session types with or without per-user Layer 3 tunnels
- **HNB-GW**: HNB Session over IuH
- **HNB-GW**: HNB-CN Session over IuPS and IuCS
- **HNB-GW**: SeGW Session IPsec Tunnel
- HSGW services for IPv4
- IPSG-only systems
- LNS session types
- PDSN services supporting simple IP, Mobile IP, and Proxy Mobile IP
- P-GW services for IPv4
- SGSN services (3G and 2.5G services) for IPv4 and PPP PDP contexts

Session recovery is **not supported** for the following functions:

- Any session using IPv6 (PDSN/GGSN/SGSN/LNS)
- Any session needing L2TP LAC support (including regenerated PPP on top of an HA/GGSN session)
- Destination-based accounting recovery
- GGSN network initiated connections
- GGSN session using more than 1 service instance
- MIP/L2TP with IPsec integration
- MIP session with multiple concurrent bindings
- Mobile IP sessions with L2TP
- Multiple MIP sessions

 **Important:** Session Recovery is a licensed Cisco feature. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of the *Software Management Operations* chapter in the *System Administration Guide*.

When session recovery occurs, the system reconstructs the following subscriber information:

- Data and control state information required to maintain correct call behavior.
- A minimal set of subscriber data statistics; required to ensure that accounting information is maintained.
- A best-effort attempt to recover various timer values such as call duration, absolute time, and others.
- The idle time timer is reset to zero and the re-registration timer is reset to its maximum value for HA sessions to provide a more conservative approach to session recovery.

 **Important:** Any partially connected calls (for example, a session where HA authentication was pending but has not yet been acknowledged by the AAA server) are not recovered when a failure occurs.

Additional Hardware Requirements

Because session recovery requires numerous hardware resources, such as memory, control processors, NPU processing capacity, etc., some additional hardware may be required to ensure that enough resources are available to fully support this feature.



Important: A minimum of four packet processing cards (three active and one standby) per individual chassis is required to use this feature.

To allow for complete session recovery in the event of a hardware failure during a packet processing card migration, a minimum of three active packet processing cards and two standby packet processing cards should be deployed.

To assist you in your network design and capacity planning, consider the following factors:

- Subscriber capacity is decreased depending on the hardware configuration. A fully configured chassis would experience a smaller decrease in subscriber capacity versus a minimally configured chassis.
- The amount by which control transaction processing capacity is reduced.
- The reduction in subscriber data throughput.
- The recovery time for a failed software task.
- The recovery time for a failed packet processing card.

A packet processing card migration may temporarily impact session recovery as hardware resources (memory, processors, etc.) that may be needed are not available during the migration. To avoid this condition, a minimum of two standby packet processing cards should be configured.

Configuring the System to Support Session Recovery

The following procedures allow you to configure the session recovery feature for either an operational system that is currently in-service (able to accept incoming calls) or a system that is out-of-service (not part of your production network and, therefore, not processing any live subscriber/customer data).



Important: The session recovery feature, even when the feature use key is present, is disabled by default on the system.

Enabling Session Recovery

As noted earlier, session recovery can be enabled on a system that is out-of-service (OOS) and does not yet have any contexts configured, or on an in-service system that is currently capable of processing calls. However, if the system is in-service, it must be restarted before the session recovery feature takes effect.

Enabling Session Recovery on an Out-of-Service System

The following procedure is for a system that does not have any contexts configured.

To enable the session recovery feature on an out-of-service system, follow the procedure below. This procedure assumes that you begin at the Exec mode prompt.

- Step 1** At the Exec mode prompt, verify that the session recovery feature is enabled by the session and feature use license on the system by running the **show license info** command:
- The output of this command appears similar to the example shown below. Note that the session recovery feature is bold-faced in this example.

```
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
```

■ Configuring the System to Support Session Recovery

```

-----
xxx-xx-xxxx      15  PDSN/GGSN/SGSN (10K)

      [none]      -   FA

      [none]      -   IPv4 Routing Protocols

xxx-xx-xxxx      -   IPSec

xxx-xx-xxxx      -   2TP LAC (PDSN/GGSN/SGSN)

xxx-xx-xxxx      1   L2TP LNS (10K)

xxx-xx-xxxx      6   L2TP LNS (1K)

xxx-xx-xxxx      -   Session Recovery (PDIF/PDSN/GGSN/SGSN)

      [none]      -   Session Recovery (HA)

xxx-xx-xxxx      -   PCF Monitoring

xxx-xx-xxxx      -   Layer 2 Traffic Management

Session Limits:

              Sessions  Session Type
              -----
              150000    PDSN/GGSN/SGSN

Status:

              16000    L2TP LNS

CF Device 1          Does not match either SPC

CF Device 2          Does not match either SPC

License Status       Good (Not Redundant)

```



Important: If the current status of the Session Recovery feature is Disabled, you cannot enable this feature until a license key is installed in the system.

Step 2 Use the following configuration example to enable session recovery.

```

configure

  require session recovery

end

```

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

The system, when started, enables session recovery, creates all mirrored “standby-mode” tasks, and performs packet processing card reservations and other operations automatically.

- Step 4** After the system has been configured and placed in-service, you should verify the preparedness of the system to support this feature as described in the *Viewing Session Recovery Status* section.

Enabling Session Recovery on an In-Service System

When enabling session recovery on a system that already has a saved configuration, the session recovery commands are automatically placed before any service configuration commands in the configuration file.

To enable the session recovery feature on an in-service system, follow the procedure below. This procedure assumes that you begin at the Exec mode prompt.

- Step 1** At the Exec mode prompt, verify that the session recovery feature is enabled by the session and feature use license on the system by running the **show license info** command:

The output of this command appears similar to the example shown below. Note that the session recovery feature is bold-faced in this example.

```
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/GGSN/SGSN (10K)
      [none]    -    FA
      [none]    -    IPv4 Routing Protocols
xxx-xx-xxxx  -    IPSec
xxx-xx-xxxx  -    2TP LAC (PDSN/GGSN/SGSN)
xxx-xx-xxxx  1    L2TP LNS (10K)
```

■ Configuring the System to Support Session Recovery

```

xxx-xx-xxxx      6    L2TP LNS (1K)

xxx-xx-xxxx      -    Session Recovery (PDIF/PDSN/GGSN/SGSN)

[none]           -    Session Recovery (HA)

xxx-xx-xxxx      -    PCF Monitoring

xxx-xx-xxxx      -    Layer 2 Traffic Management

Session Limits:

                Sessions  Session Type
                -----  -
                150000    PDSN/GGSN/SGSN

Status:

                16000    L2TP LNS

CF Device 1      Does not match either SPC

CF Device 2      Does not match either SPC

License Status    Good (Not Redundant)

```



Important: If the current status of the Session Recovery feature is Disabled, you cannot enable this feature until a license key is installed in the system.

Step 2 Use the following configuration example to enable session recovery.

```

configure

  require session recovery

end

```



Important: This feature does not take effect until after the system has been restarted.

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

Step 4 Perform a system restart by entering the **reload** command:

The following prompt appears:

Are you sure? [Yes|No]:

Confirm your desire to perform a system restart by entering **yes**.

The system, when restarted, enables session recovery and creates all mirrored “standby-mode” tasks, performs packet processing card reservations, and other operations automatically.

- Step 5** After the system has been restarted, you should verify the preparedness of the system to support this feature as described in the *Viewing Session Recovery Status* section.



Important: More advanced users may opt to simply insert the **require session recovery** command syntax into an existing configuration file using a text editor or other means, and then applying the configuration file manually. Exercise caution when doing this to ensure that this command is placed among the first few lines of any existing configuration file; it must appear before the creation of any non-local context.

Disabling the Session Recovery Feature

To disable the session recovery feature on a system, enter the **no require session recovery** command from the Global Configuration mode prompt.



Important: If this command is issued on an in-service system, then the system must be restarted by issuing the **reload** command.

Viewing Session Recovery Status

To determine if the system is capable of performing session recovery, when enabled, enter the **show session recovery status verbose** command from the Exec mode prompt.

The output of this command should be similar to the examples shown below.

```
[local]host_name# show session recovery status

Session Recovery Status:

Overall Status           : SESSMGR Not Ready For Recovery

Last Status Update      : 1 second ago
```

```
[local]host_name# show session recovery status

Session Recovery Status:

Overall Status           : Ready For Recovery

Last Status Update      : 8 seconds ago
```

```
[local]host_name# show session recovery status verbose

Session Recovery Status:

Overall Status           : Ready For Recovery

Last Status Update      : 2 seconds ago
```

■ Configuring the System to Support Session Recovery

```

-----sessmgr-----aaamgr-----demux
cpu state  active  standby  active  standby  active  status
-----
1/1 Active  2      1      1      1      0      Good
1/2 Active  1      1      0      0      0      Good
1/3 Active  1      1      3      1      0      Good
2/1 Active  1      1      1      1      0      Good
2/2 Active  1      1      0      0      0      Good
2/3 Active  2      1      3      1      0      Good
3/0 Active  0      0      0      0      1      Good (Demux)
3/2 Active  0      0      0      0      1      Good (Demux)
4/1 Standby 0      2      0      1      0      Good
4/2 Standby 0      1      0      0      0      Good
4/3 Standby 0      2      0      3      0      Good

[local]host_name#

```

Viewing Recovered Session Information

To view session state information and any session recovery status, enter the following command:

```
show subscriber debug-info { callid id | msid id | username name }
```

Keyword/Variable	Description
callid <i>id</i>	Displays subscriber information for the call specified by <i>id</i> . The call ID is an 8-byte hexadecimal number.
msid <i>id</i>	Displays information for the mobile user identified by an MSID from 7 to 16 digits specified as an IMSI, MIN, or RMI. Wildcard characters \$ and * are allowed. The * wildcard matches multiple characters and the \$ wildcard matches a single character. If you do not want the wildcard characters interpreted as a wildcard enclose them in single quotes ('). For example; '\$'.
username <i>name</i>	Displays information for connections for the subscriber identified by a previously configured username. <i>name</i> is a sequence of alphanumeric characters and/or wildcard characters ('\$ and '*') from 1 to 127 characters in length. The * wildcard matches multiple characters and the \$ wildcard matches a single character. If you do not want the wildcard characters interpreted as wildcard enclose them in single quotes ('). For example; '\$'.

The following example shows the output of this command both before and after a session recovery operation has been performed. The “Redundancy Status” fields in this example have been bold-faced for clarity.

```

username: user1          callid: 01ca11b1          msid: 0000100003

Card/Cpu: 4/2

Sessmgr Instance: 7

Primary callline:

Redundancy Status: Original Session

Checkpoints    Attempts    Success    Last-Attempt    Last-Success

  Full:           69         68         29800ms         29800ms

  Micro:          206        206         20100ms         20100ms

Current state: SMGR_STATE_CONNECTED

FSM Event trace:

      State                               Event

SMGR_STATE_OPEN                          SMGR_EVT_NEWCALL

SMGR_STATE_NEWCALL_ARRIVED               SMGR_EVT_ANSWER_CALL

SMGR_STATE_NEWCALL_ANSWERED              SMGR_EVT_LINE_CONNECTED

SMGR_STATE_LINE_CONNECTED                 SMGR_EVT_LINK_CONTROL_UP

SMGR_STATE_LINE_CONNECTED                 SMGR_EVT_AUTH_REQ

SMGR_STATE_LINE_CONNECTED                 SMGR_EVT_IPADDR_ALLOC_SUCCESS

SMGR_STATE_LINE_CONNECTED                 SMGR_EVT_AUTH_SUCCESS

SMGR_STATE_LINE_CONNECTED                 SMGR_EVT_UPDATE_SESS_CONFIG

SMGR_STATE_LINE_CONNECTED                 SMGR_EVT_LOWER_LAYER_UP

Data Reorder statistics

Total timer expiry:                      0          Total flush (tmr expiry):    0

Total no buffers:                        0          Total flush (no buffers):    0

Total flush (queue full):                 0          Total flush (out of range):  0

Total flush (svc change):                 0          Total out-of-seq pkt drop:   0

Total out-of-seq arrived:                 0

```

IPv4 Reassembly Statistics:

```

Success:                0          In Progress: 0
Failure (timeout):      0          Failure (no buffers): 0
Failure (other reasons): 0

```

Redirected Session Entries:

```

Allowed:                2000       Current:                0
Added:                  0          Deleted:                0
Revoked for use by different subscriber: 0

```

Peer callline:

Redundancy Status: Original Session

Checkpoints	Attempts	Success	Last-Attempt	Last-Success
Full:	0	0	0ms	0ms
Micro:	0	0	0ms	0ms

Current state: SMGR_STATE_CONNECTED

FSM Event trace:

State	Event
SMGR_STATE_LINE_CONNECTED	SMGR_EVT_LOWER_LAYER_UP
SMGR_STATE_CONNECTED	SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED	SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED	SMGR_EVT_REQ_SUB_SESSION
SMGR_STATE_CONNECTED	SMGR_EVT_RSP_SUB_SESSION
SMGR_STATE_CONNECTED	SMGR_EVT_ADD_SUB_SESSION
SMGR_STATE_CONNECTED	SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED	SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED	SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED	SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED	SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED	SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED	SMGR_EVT_AUTH_REQ


```

SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS

SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_REQ

SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS

Data Reorder statistics

Total timer expiry:           0          Total flush (tmr expiry): 0
Total no buffers:              0          Total flush (no buffers): 0
Total flush (queue full):      0          Total flush (out of range):0
Total flush (svc change):      0          Total out-of-seq pkt drop: 0
Total out-of-seq arrived:      0

IPv4 Reassembly Statistics:

Success:                       0          In Progress:                0
Failure (timeout):              0          Failure (no buffers):        0
Failure (other reasons):        0

Redirected Session Entries:

Allowed:                        2000       Current:                     0
Added:                          0          Deleted:                     0

Revoked for use by different subscriber: 0

```

Notice that in the example above, where the session has been recovered/recreated, that state events (FSM Event State field) no longer exist. This field is re-populated as new state changes occur.

Appendix J

VLANs

This appendix provides information on configuring an enhanced, or extended, service. The product administration guides provide examples and procedures for configuration of basic services on the system. You should select the configuration example that best meets your service model before using the procedures in this appendix.

This appendix includes the following sections:

- [Overview](#)
- [Creating VLAN Tags](#)
- [Configuring Subscriber VLAN Associations](#)

Overview

Virtual LANs (VLANs) provide greater flexibility in the configuration and use of contexts and services.

They are configured as “tags” on a per-port basis and allow more complex configurations to be implemented. The VLAN tag allows a single physical port to be bound to multiple logical interfaces that can be configured in different contexts. Therefore, each Ethernet port can be viewed as containing many logical ports when VLAN tags are employed.



Important: VLANs are supported in conjunction with subscriber traffic ports on Ethernet line cards. The system supports the configuration limits for VLANs as described in the *Engineering Rules* appendix of this guide.

Creating VLAN Tags

Use the following example to create VLANs on a port and bind them to pre-existing interfaces. For information on creating interfaces, refer to the *System Element Configuration Procedures* chapter of this guide.

config

```
port ethernet <slot/port>

  no shutdown

  vlan <vlan_tag_ID>

  no shutdown

  bind interface <interface_name> <context_name>

end
```

Notes:

- *Optional:* Configure VLAN-subscriber associations. Refer to the *Configuring Subscriber VLAN Associations* section for more information.
- Repeat this procedure as needed to configure additional VLANs for the port.

Verify the Port Configuration – ASR 5000

Run the following command to verify the port configuration:

```
show port info slot/port
```

An example of this command's output is shown below:

```
Port: 17/1

Port Type : 10/100 Ethernet

Description : (None Set)

Controlled By Card : 1 (Packet Accelerator Card)

Redundancy Mode : Card Mode

Redundant With : 33/1

Physical ifIndex : 285278208

Administrative State : Enabled

Configured Duplex : Auto

Configured Speed : Auto

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

Link State : Up

Link Duplex : Unknown

Link Speed : Unknown

Untagged:

Logical ifIndex : 285278209

Operational State : Down, Active

Tagged VLAN: VID 10

Logical ifIndex : 285278210

VLAN Type : Subscriber

Administrative State : Enabled

Operational State : Up, Active

Number of VLANs : 1
```

Notes:

- *Optional:* Repeat this configuration as needed to configure additional ports.
- *Optional:* Configure VLAN-subscriber associations if needed.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

Configuring Subscriber VLAN Associations

Subscriber traffic can be routed to specific VLANs based on the configuration of their user profile. This functionality provides a mechanism for routing all traffic from a subscriber over the specified VLAN. All packets destined for the subscriber must also be sent using only IP addresses valid on the VLAN or they will be dropped.

RADIUS Attributes Used

The following RADIUS attributes can be configured within subscriber profiles on the RADIUS server to allow the association of a specific VLAN to the subscriber:

- **SN-Assigned-VLAN-ID:** In the Starent VSA dictionary
- **SN1-Assigned-VLAN-ID:** In the Starent VSA1 dictionary



Important: Since the instructions for configuring subscriber profiles differ between RADIUS server applications, this section only describes the individual attributes that can be added to the subscriber profile. Please refer to the documentation that shipped with your RADIUS server for instructions on configuring subscribers.

Configuring Local Subscriber Profiles

Use the configuration example below to configure VLAN associations within local subscriber profiles on the system.



Important: These instructions assume that you have already configured subscriber-type VLAN tags according to the instructions provided in the *Creating VLAN Tags* section of this appendix.

```
config

  context <context_name>

    subscriber name <user_name>

      ip vlan <vlan_id>

    end
```

Verify the Subscriber Profile Configuration

Use the following command to view the configuration for a subscriber profile:

```
show subscriber configuration username user_name
```

Notes:

- Repeat this command for each additional subscriber.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter of this guide.

