



Subscriber Information Collector (SIC)



Modified: 2016-12-29

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Copyright © 2017 Juniper Networks, Inc. All rights reserved.

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. and/or its affiliates in the United States and other countries. All other trademarks may be property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Subscriber Information Collector (SIC)

Copyright © 2017 Juniper Networks, Inc. All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula/>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

	About the Documentation	xiii
	Documentation and Release Notes	xiii
	Supported Platforms	xiii
	Documentation Conventions	xiii
	Documentation Conventions	xiv
	Documentation Feedback	xvi
	Requesting Technical Support	xvi
	Self-Help Online Tools and Resources	xvii
	Opening a Case with JTAC	xvii
Part 1	Overview	
Chapter 1	Software Features Overview	3
	SRC Component Overview	3
Chapter 2	Subscriber Information Collector	7
	Subscriber Information Collector Overview	7
	RADIUS Authentication/Authorization and Accounting Data Flow	8
	COA Authentication Data Flow	10
	COA Accounting Data Flow	11
	SIC Accounting Data Flow (Accounting Target=Proxy)	11
	Local and Shared Configurations for the SIC (SRC CLI)	12
	Local Configuration	12
	Shared Configuration	12
	Accounting Method and Target (SRC CLI)	13
	Using the Proxy RADIUS Accounting Method	13
	Authentication Route Targets (SRC CLI)	14
	Request Routing (SRC CLI)	14
	Explicit Routing Rules	15
	Implicit Routing Rules	16
	SIC Editing Rules (SRC CLI)	17
	RADIUS and Diameter Configuration for the SIC Overview (SRC CLI)	21
	RADIUS Accounting and Authentication Listeners	22
	SIC Diameter Server	22
	SIC Diameter Server Configuration Overview	23
	RADIUS Network Elements	23
	Configuring Upstream RADIUS Network Elements Overview	24
	Configuring Downstream RADIUS Network Elements Overview	25
	Using the Proxy Function to Define Implicit Routing Rules	25

	Failover Policy	25
	Failover Mode	26
	Round-Robin	26
	Primary or Backup	27
	RADIUS and Diameter Transports	27
	Inbound and Outbound RADIUS Accounting Transports for the SIC Group	27
	Inbound and Outbound RADIUS Authentication Transports for the SIC Group	27
	SIC Server Inbound and Outbound RADIUS Transports	27
	Diameter Transport	27
	SIC Dictionaries and Device Models Overview (SRC CLI)	28
	Dictionaries and the Device Models Supported by the SIC Group	28
	Configuring Device Models and Their Associated Dictionaries for the SIC Group Overview	28
	Modifying a Dictionary	29
	Configuring the Dictionaries Used by the SIC Group Overview	29
	SIC Local Realms Overview	29
	SIC Event Logging Overview (SRC CLI)	29
	Log File Options	30
	Event Levels	31
	Log Groups	32
	SNMP Support for the SIC Overview (SRC CLI)	32
Chapter 3	RADIUS COA	35
	Managing Dynamic Services on RADIUS-Enabled Devices	35
	SIC Dynamic Authorization Support Overview	36
	Rendering	37
	How the Dynamic Authorization Process Works in the SIC	38
	Introduction	38
	Initial Authorization	38
	Accounting	39
	Service Activation and Deactivation	40
	Abort Session Requests	41
	Dynamic Authorization Targets (SRC CLI)	42
Part 2	Configuration	
Chapter 4	Configuration Tasks for Subscriber Information Collector	45
	SIC Configuration Summary	45
	SIC RADIUS Configuration Summary (SRC CLI)	46
	SIC RADIUS Dynamic Authorization Configuration Summary (SRC CLI)	47
	SIC Diameter Configuration Summary (SRC CLI)	48
	Configuring Management of RADIUS-Enabled Devices for the SIC (SRC CLI)	48
	Configuring the Connection Between the SIC and the Juniper Networks Database (SRC CLI)	48
	Creating an SIC Group and Server (SRC CLI)	50
	Creating an SIC Server Instance (SRC CLI)	51
	Configuring Dictionaries for the SIC Group (SRC CLI)	51

Configuring the Device Models Supported by the SIC Group (SRC CLI)	54
Configuring the RADIUS Accounting Listener for the SIC Group (SRC CLI)	54
Configuring the RADIUS Accounting Listener Queue Limits (SRC CLI)	54
Configuring the RADIUS Accounting Listener Transport (SRC CLI)	55
Configuring the RADIUS Authentication Listener for the SIC Group (SRC CLI)	56
Configuring the RADIUS Authentication Listener Queue Limits (SRC CLI)	56
Configuring the RADIUS Authentication Listener Transport (SRC CLI)	57
Configuring the Outbound RADIUS Transport of the SIC Group (SRC CLI)	58
Configuring the RADIUS Transport for an SIC Server (SRC CLI)	59
Configuring the SIC Diameter Server (SRC CLI)	60
Configuration Statements for the SIC Diameter Server (SRC CLI)	60
Configuring the SIC Diameter Server Identity (SRC CLI)	61
Configuring the SIC Diameter Server Peer (SRC CLI)	62
Configuring Upstream and Downstream RADIUS Network Elements (SRC CLI)	64
Configuration Statements for Downstream Network Elements and Accounting and Authentication Targets (SRC CLI)	65
Configuration Statements for Upstream Network Elements, Accounting and Authentication Clients, and Dynamic Authorization Targets (SRC CLI)	66
Creating a Network Element (SRC CLI)	66
Configuring the Device Models Supported in the Network Element (SRC CLI)	67
Configuring Upstream Network Elements and Accounting and Authentication Clients (SRC CLI)	68
Configuring Upstream Network Elements and Dynamic Authorization Targets (SRC CLI)	69
Configuring Downstream Network Elements and Accounting and Authentication Targets (SRC CLI)	70
Configuring SIC Accounting Targets (SRC CLI)	70
Configuring SIC Authentication Targets (SRC CLI)	71
Configuration Statements for SIC Group Failover Mode and Policy (SRC CLI)	72
Configuring Failover Mode and Policy (SRC CLI)	73
Configuring Failover Mode (SRC CLI)	73
Configuring Fast Fail Options for the Failover Policy	74
Configuring Retry Options for the Failover Policy	75
Configuring What Realms Are Local to the SIC Group (SRC CLI)	75
Configuration Statements for SIC Editing Rules (SRC CLI)	76
Configuring the Optional Editing Rules Used by the SIC Group (SRC CLI)	80
Configuring the Accounting Method Used by the SIC Group (SRC CLI)	82
Configuring Proxy RADIUS as the Accounting Method (SRC CLI)	82
Configuring the Authentication Target Used by the SIC Server (SRC CLI)	82
Configuring Request Routing (SRC CLI)	83
Configuration Statements for SIC Explicit Accounting Routing Rules	83
Configuration Statements for SIC Explicit Authentication Routing Rules	84
Configuring Explicit Routing (SRC CLI)	85
Configuring Implicit Routing (SRC CLI)	88
Configuring Event Logging for an SIC Server (SRC CLI)	88

Chapter 5

Configuring SNMP for the SIC Group (SRC CLI)	92
Configuration Tasks for RADIUS COA	95
Device and Service Template Configuration Overview (SRC CLI)	95
Device Template Configuration Overview (SRC CLI)	96
Service and Global Service Template Configuration Overview (SRC CLI)	96
Mode	97
Attributes	98
Variables	100
Tagged Attributes	101
SIC RADIUS Dynamic Authorization Configuration Summary (SRC CLI)	101
Configuring Device Templates (SRC CLI)	102
Configuring the Device Capabilities Supported in the Device Template (SRC CLI)	103
Configuration Statements for SIC Service Templates (SRC CLI)	104
Configuring SIC Service Templates (SRC CLI)	105
Creating an SIC Service Template (SRC CLI)	106
Configuring the Mode of the SIC Service Template (SRC CLI)	106
Configuring Variables for the SIC Service Template (SRC CLI)	107
Configuring Normal Attributes for the SIC Service Template (SRC CLI)	107
Configuring Required Attributes for the SIC Service Template (SRC CLI)	109
Configuring Default Attributes for the SIC Service Template (SRC CLI)	110
Configuring Parameterized Attributes for the SIC Service Template (SRC CLI)	111
Configuring Override Attributes for the SIC Service Template (SRC CLI)	112
Configuration Statements for Tagged Attributes in SIC Service Templates (SRC CLI)	113
Configuring Tagged Attributes in SIC Service Templates (SRC CLI)	114
Creating a Tagged Attribute Group in the SIC Service Template (SRC CLI)	115
Configuring Normal Attributes in a Tagged Attribute Group (SRC CLI)	115
Configuring Default Attributes in a Tagged Attribute Group (SRC CLI)	116
Configuring Required Attributes in a Tagged Attribute Group (SRC CLI)	118
Configuring Override Attributes in a Tagged Attribute Group (SRC CLI)	119
Configuring Parameterized Attributes in a Tagged Attribute Group (SRC CLI)	120
Configuration Statements for SIC Global Service Templates (SRC CLI)	121
Configuring Global Service Templates (SRC CLI)	122
Creating an SIC Global Service Template (SRC CLI)	123
Configuring the Mode of the SIC Global Service Template (SRC CLI)	123
Configuring Variables for the SIC Global Service Template (SRC CLI)	123
Configuring Normal Attributes for the SIC Global Service Template (SRC CLI)	124
Configuring Required Attributes for the SIC Global Service Template (SRC CLI)	125
Configuring Default Attributes for the SIC Global Service Template (SRC CLI)	126
Configuring Parameterized Attributes for the SIC Global Service Template (SRC CLI)	128

	Configuring Override Attributes for the SIC Global Service Template (SRC CLI)	129
	Configuring Management of RADIUS-Enabled Devices for the SIC (SRC CLI)	130
	Configuring Upstream Network Elements and Dynamic Authorization Targets (SRC CLI)	131
	SIC Diameter Configuration Summary (SRC CLI)	132
	Configuring the SIC Diameter Server (SRC CLI)	133
	Configuration Statements for the SIC Diameter Server (SRC CLI)	133
	Configuring the SIC Diameter Server Identity (SRC CLI)	134
	Configuring the SIC Diameter Server Peer (SRC CLI)	135
	Configuring the Diameter Application (SRC CLI)	137
	Configuring the Diameter Application Properties	138
	Configuring the Diameter Client Properties	141
	Configuring the Diameter Server Properties	142
	Configuring Logging Destinations	142
	Configuring Diameter Peers (SRC CLI)	143
	Configuring the NAS Groups (SRC CLI)	145
	Configuring NAS Groups	145
	Configuring the NAS Group Device Capabilities (SRC CLI)	146
	Classifying Interfaces	147
	Configuring NAS Group Routes	148
	Configuring the SAE to Manage AAA Devices	150
	Configuring AAA Policies (SRC CLI)	152
	Configuring AAA Policy Lists	152
	Configuring AAA Policy Rules	152
	Configuring Template Activation Actions	152
Chapter 6	Example for Subscriber Information Collector	155
	Example: Basic SIC Group Configuration (SRC CLI)	155
	Sample Service Templates	165
	Juniper Networks Routers Service Template	165
	Cisco Router Service Template	170
Part 3	Administration	
Chapter 7	Routine Monitoring	179
	Viewing Statistics for RADIUS Client Accounting Requests (SRC CLI)	179
	Viewing Statistics for RADIUS Client Authentication Requests (SRC CLI)	179
	Viewing RADIUS Host Statistics for Accounting Transactions (SRC CLI)	180
	Viewing RADIUS Host Statistics for Authentication Transactions (SRC CLI)	180
	Viewing RADIUS Target Statistics for Accounting Requests (SRC CLI)	181
	Viewing RADIUS Target Statistics for Authentication Requests (SRC CLI)	181
	Viewing Diameter Host Statistics (SRC CLI)	182
	Viewing Diameter Peer Statistics (SRC CLI)	182

List of Figures

Part 1	Overview	
Chapter 2	Subscriber Information Collector	7
	Figure 1: Data Flow	9
	Figure 2: Explicit Routing Rule Process	15
	Figure 3: SIC Editing Rule Process	17
	Figure 4: Editing and Accounting Routing Rule Conditions and Processes	20
	Figure 5: SIC RADIUS and Diameter Configuration	22
	Figure 6: SIC Diameter Configuration	23
	Figure 7: Upstream and Downstream Network Element Clients and Targets	24
	Figure 8: Upstream and Downstream Network Element Client and Target Configuration Options	24
	Figure 9: Round-Robin	26
	Figure 10: SNMP Support for the SIC	33
Chapter 3	RADIUS COA	35
	Figure 11: The Rendering Process	37
	Figure 12: Initial Authorization and Accounting Timing Sequence	40
	Figure 13: Activation and Deactivation Timing Sequences	41
	Figure 14: Abort Session Timing Sequence	42

List of Tables

	About the Documentation	xiii
	Table 1: Notice Icons	xiv
	Table 2: Notice Icons	xv
	Table 3: Text Conventions	xv
Part 1	Overview	
Chapter 1	Software Features Overview	3
	Table 4: Descriptions of SRC Components	3
Chapter 2	Subscriber Information Collector	7
	Table 5: SIC Log File Options	30
	Table 6: SIC Event Levels	31
	Table 7: SIC Log Groups	32
	Table 8: MIBs Used by the SIC for SNMP Statistics	33
	Table 9: SNMP Traps Supported for the SIC	33
Part 2	Configuration	
Chapter 4	Configuration Tasks for Subscriber Information Collector	45
	Table 10: SIC Editing Rule Options	80
	Table 11: Explicit Routing Rule Conditions	86
Chapter 5	Configuration Tasks for RADIUS COA	95
	Table 12: Device Template Capabilities and Associated Values	96
	Table 13: Service Template Modes	97
	Table 14: Global Service Template Modes	97
	Table 15: Attributes for All Modes	98
	Table 16: Variables	100
	Table 17: Capabilities and Associated Values	103
Chapter 6	Example for Subscriber Information Collector	155
	Table 18: Sample Configuration Attribute Associations	155
	Table 19: Log Groups and Associated Event Level for Log Stream=default logger	156
	Table 20: Log Groups and Associated Event Level for Log Stream=debug-logger	156
	Table 21: Log Groups and Associated Event Level for Log Stream=error-logger	157

About the Documentation

- Documentation and Release Notes on page xiii
- Supported Platforms on page xiii
- Documentation Conventions on page xiii
- Documentation Feedback on page xvi
- Requesting Technical Support on page xvi

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at <http://www.juniper.net/techpubs/>.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at <http://www.juniper.net/books>.

Supported Platforms







For the features described in this document, the following platforms are supported:

- Virtualized SRC

Documentation Conventions

Table 1 on page xiv defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Documentation Conventions

[Table 1 on page xiv](#) defines the notice icons used in this guide. [Table 3 on page xv](#) defines text conventions used throughout this documentation.

Table 2: Notice Icons







Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 3: Text Conventions

Convention	Description	Examples
Bold text like this	<ul style="list-style-type: none"> Represents keywords, scripts, and tools in text. Represents a GUI element that the user selects, clicks, checks, or clears. 	<ul style="list-style-type: none"> Specify the keyword exp-msg. Run the install.sh script. Use the pkgadd tool. To cancel the configuration, click Cancel.
Bold text like this	Represents text that the user must type.	user@host# set cache-entry-age <i>cache-entry-age</i>
Fixed-width text like this	Represents information as displayed on your terminal's screen, such as CLI commands in output displays.	<pre> nic-locators { login { resolution { resolver-name /realms/ login/A1; key-type LoginName; value-type SaeId; } } } </pre>
Regular sans serif typeface	<ul style="list-style-type: none"> Represents configuration statements. Indicates SRC CLI commands and options in text. Represents examples in procedures. Represents URLs. 	<ul style="list-style-type: none"> system ldap server{ stand-alone; Use the request sae modify device failover command with the force option user@host# ... http://www.juniper.net/techpubs/software/management/sdx/api-index.html

Table 3: Text Conventions (*continued*)

<i>Italic sans serif typeface</i>	Represents variables in SRC CLI commands.	<code>user@host# set local-address local-address</code>
Angle brackets	In text descriptions, indicate optional keywords or variables.	Another runtime variable is <gfwif>.
Key name	Indicates the name of a key on the keyboard.	Press Enter.
Key names linked with a plus sign (+)	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
<i>Italic typeface</i>	<ul style="list-style-type: none"> Emphasizes words. Identifies book names. Identifies distinguished names. Identifies files, directories, and paths in text but not in command examples. 	<ul style="list-style-type: none"> There are two levels of access: <i>user</i> and <i>privileged</i>. <i>SRC-PE Getting Started Guide</i>. <i>o=Users, o=UMC</i> The <i>/etc/default.properties</i> file.
Backslash	At the end of a line, indicates that the text wraps to the next line.	<code>Plugin.radiusAcct-1.class=\ net.juniper.smgmt.sae.plugin\ RadiusTrackingPluginEvent</code>
Words separated by the symbol	Represent a choice to select one keyword or variable to the left or right of this symbol. (The keyword or variable may be either optional or required.)	<code>diagnostic line</code>

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at <http://www.juniper.net/techpubs/index.html>, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at <http://www.juniper.net/techpubs/feedback/>.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes: <http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <http://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum: <http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://tools.juniper.net/SerialNumberEntitlementSearch/>

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/>.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see <http://www.juniper.net/support/requesting-support.html>.

PART 1

Overview

- [Software Features Overview on page 3](#)
- [Subscriber Information Collector on page 7](#)
- [RADIUS COA on page 35](#)

CHAPTER 1

Software Features Overview

- [SRC Component Overview on page 3](#)

SRC Component Overview

The SRC software is a dynamic system. It contains many components that you use to build a subscriber management environment. You can use these tools to customize and extend the SRC software for your use and to integrate the SRC software with other systems. The SRC software also provides the operating system and management tools for C Series Controllers.

[Table 4 on page 3](#) gives a brief description of the components that make up the SRC software.

Table 4: Descriptions of SRC Components

Component	Description
Server Components	
Service activation engine (SAE)	<ul style="list-style-type: none">• Authorizes, activates, and deactivates subscriber and service sessions by interacting with systems such as Juniper Networks routers, cable modem termination system (CMTS) devices, RADIUS servers, and directories.• Collects accounting information about subscribers and services from routers, and stores the information in RADIUS accounting servers, flat files, and other accounting databases.• Provides plug-ins and application programming interfaces (APIs) for starting and stopping subscriber and service sessions and for integrating with systems that authorize subscriber actions and track resource usage.
Subscriber Information Collector (SIC)	The SIC listens for RADIUS accounting events from IP edge devices (accounting clients) and forwards them to a remote AAA server, allowing the SRC software to gain increased subscriber awareness. Additionally, the SIC can optionally edit accounting events before routing them.
Juniper Policy Server (JPS)	Acts as a policy decision point (PDP) and policy enforcement point (PEP) that manages the relationships between application managers and CMTS devices in a PCMM environment.
Network information collector (NIC)	Collects information about the state of the network and can provide a mapping from a given type of network data to another type of network data.
Redirect Server	Redirects HTTP requests received from IP Filter to a captive portal page.

Table 4: Descriptions of SRC Components *(continued)*

Component	Description
3GPP Gateway	The SRC Third-Generation Partnership Project (3GPP) gateway is a Diameter-based component in the SRC software, which provides integration with 3GPP Policy and Charging Control environments, to provide fixed-mobile convergence (FMC). The SRC 3GPP gateway provides Gx-based integration with the Policy and Charging Rules Function (PCRF). The SRC 3GPP gateway uses the northbound Gx interface to mediate between the PCRF and Juniper Networks routers like the E Series Broadband Services routers and MX Series routers. The northbound Gx interface on the SRC 3GPP gateway communicates with the PCRF using the Diameter protocol.
3GPP Gy	The SRC 3GPP Gy is a Diameter-based component in the SRC software, which provides Gy-based integration with the Online Charging System (OCS), to provide FMC. The SRC 3GPP Gy uses the northbound Gy interface to handle charging-related information between the OCS and Juniper Networks routers like the E Series Broadband Services routers and MX Series routers. The northbound Gy interface communicates with the OCS using the Diameter protocol.
Web Application Service	The SRC software includes a Web application server that hosts the Web Services Gateway and the Volume Tracking Application (SRC VTA). In production environments, this application server is designed to host only these applications. However, you can load your own applications into this server for testing or demonstration purposes.
Web Services Gateway	<p>Allows a gateway client—an application that is not part of the SRC network—to interact with SRC components through a Simple Object Access Protocol (SOAP) interface.</p> <p>The Web Services Gateway provides the Dynamic Service Activator which allows a gateway client to dynamically activate and deactivate SRC services for subscribers and to run scripts that manage the SAE.</p>
Repository	
Directory	<p>The SRC software includes the Juniper Networks database, which is a built-in Lightweight Directory Access Protocol (LDAP) directory for storing all SRC data including services, policies, and small subscriber databases.</p> <p>For large subscriber databases, you must supply your own directory.</p>
SRC Configuration and Management Tools	
SRC command line interface (CLI)	Provides a way to configure the SRC software on a C Series Controller from a Junos OS–like CLI. The SRC CLI includes the policies, services, and subscribers CLI, which has separate access privileges.
C-Web interface	Provides a way to configure, monitor, and manage the SRC software on a C Series Controller through a Web browser. The C-Web interface includes a policies, services, and subscribers component, which has separate access privileges.
Simple Network Management Protocol (SNMP) agent	Monitors system performance and availability. It runs on all the SRC hosts and makes management information available through SNMP tables and sends notifications by means of SNMP traps.
Service Management Applications (Run on external system)	
IMS Services Gateway	Integrates into an IP multimedia system (IMS) environment. The SRC software provides a Diameter protocol-based interface that allows the SRC software to integrate with services found on the application layer of IMS.

Table 4: Descriptions of SRC Components *(continued)*

Component	Description
SRC Programming Interfaces	
NETCONF API	Allows you to configure or request information from the NETCONF server on a C Series Controller that runs the SRC software. Applications developed with the NETCONF API run on a system other than a C Series Controller.
CORBA plug-in service provider interface (SPI)	Tracks sessions and enables linking the rest of the service provider's operations support system (OSS) with the SRC software so that the OSS can be notified of events in the life cycle of SAE sessions. Hosted plug-ins only.
CORBA remote API	Provides remote access to the SAE core API. Applications that use these extensions to the SRC software run on a system other than a C Series Controller.
NIC access API	Performs NIC resolutions. Applications that use these extensions to the SRC software run on a system other than a C Series Controller.
SAE core API	Controls the behavior of the SRC software. Applications that use these extensions to the SRC software run on a system other than a C Series Controller.
Script services	Provides an interface to call scripts that supply custom services such as provisioning policies on a number of systems across a network.
VTA API	The Volume Tracking Application (VTA) API is a Simple Object Access Protocol (SOAP) interface that allows developers to create gateway clients and that administrators use to manage VTA subscribers and sessions. The SRC Web Services Gateway allows a gateway client—an application that is not part of the SRC network—to interact with SRC components, such as the VTA, through a SOAP interface.
Authorization and Accounting Applications	
AAA RADIUS servers	Authenticates subscribers and authorizes their access to the requested system or service. Accepts accounting data—time active and volume of data sent—about subscriber and service sessions. RADIUS servers run on a system other than a C Series Controller.
SRC Admission Control Plug-In (SRC ACP)	Authorizes and tracks subscribers' use of network resources associated with services that the SRC application manages.
Flat file accounting	Stores tracking data to accounting flat files that can be made available to external systems that send the data to a rating and billing system.
Volume Tracking Application	<p>The SRC Volume Tracking Application (SRC VTA) is an SRC component that allows service providers to track and control the network usage of subscribers and services. You can control volume and time usage on a per-subscriber or per-service basis. This level of control means that service providers can offer tiered services that use volume as a metric, while also controlling abusive subscribers and applications.</p> <p>When a subscriber or service exceeds bandwidth limits (or quotas), the SRC VTA can take actions including imposing rate limits on traffic, sending an e-mail notification, or charging extra for additional bandwidth consumed.</p>
Demonstration Applications (available on the Juniper Networks Website)	

Table 4: Descriptions of SRC Components *(continued)*

Component	Description
Enterprise Audit Plug-In	Defines a callback interface, which receives events when IT managers complete specified operations.
Enterprise Manager Portal	<p>Allows service providers to provision services for enterprise subscribers on routers running JunosE or Junos OS and allows IT managers to manage services.</p> <p>Enterprise Manager Portal can be used with NAT Address Management Portal to allow service providers to manage public IP addresses for use with NAT services on routers running Junos OS and to allow IT managers to make requests about public IP addresses through the Enterprise Manager Portal.</p>
Monitoring Agent application	Integrates IP address managers, such as a DHCP server or a RADIUS server, into an SRC-managed network so that the SAE is notified about subscriber events. The Monitoring Agent application runs on a Solaris platform.
Residential service selection portals	Provides a framework for building Web applications that allow residential and enterprise subscribers to manage their own network services. It comes with several full-featured sample Web applications that are easy to customize and suitable for deployment. The Residential service selection portals run on a Solaris platform.
Sample enterprise service portal	Lets service providers supply an interface to their business customers for managing and provisioning services.

Related Documentation • *SRC Product Description*

CHAPTER 2

Subscriber Information Collector

- [Subscriber Information Collector Overview on page 7](#)
- [RADIUS Authentication/Authorization and Accounting Data Flow on page 8](#)
- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 12](#)
- [Accounting Method and Target \(SRC CLI\) on page 13](#)
- [Authentication Route Targets \(SRC CLI\) on page 14](#)
- [Request Routing \(SRC CLI\) on page 14](#)
- [SIC Editing Rules \(SRC CLI\) on page 17](#)
- [RADIUS and Diameter Configuration for the SIC Overview \(SRC CLI\) on page 21](#)
- [Failover Policy on page 25](#)
- [RADIUS and Diameter Transports on page 27](#)
- [SIC Dictionaries and Device Models Overview \(SRC CLI\) on page 28](#)
- [SIC Local Realms Overview on page 29](#)
- [SIC Event Logging Overview \(SRC CLI\) on page 29](#)
- [SNMP Support for the SIC Overview \(SRC CLI\) on page 32](#)

Subscriber Information Collector Overview

The subscriber information collector (SIC) is used to manage the dynamic services on RADIUS-enabled devices. To manage dynamic services on RADIUS-enabled devices, the SIC converts the RADIUS request to Diameter request and vice versa. For information about the SIC dynamic authorization support and managing the dynamic services, see [“SIC Dynamic Authorization Support Overview” on page 36](#) and [“Managing Dynamic Services on RADIUS-Enabled Devices” on page 35](#). The SIC listens for RADIUS accounting and authentication messages from IP edge devices (clients), either directly or indirectly through an authentication, authorization, and accounting (AAA) proxy server, allowing the SRC software to gain increased subscriber awareness.

The role of the SIC is to listen for RADIUS accounting and authentication messages and filter undesired events based on attachment session attributes. The SIC is also responsible for sending RADIUS requests to the correct SRC that is managing the MX Series router.

The major components of the SIC are:

- RADIUS accounting and authentication listeners, which are configured with port numbers and parameters controlling receipt of UDP packets.
- A collection of RADIUS dictionaries.
- A collection of network access server (NAS) accounting and authentication clients.
- A collection of RADIUS accounting, authentication, and dynamic authorization targets.
- A collection of accounting and authentication routing rules.
- A collection of editing rules.
- A collection of RADIUS network elements. A RADIUS network element contains an ordered list of RADIUS accounting and authentication clients, RADIUS accounting, authentication, and dynamic authorization targets, along with a failover policy for targets.
- A proxy accounting method that forwards accounting events to a downstream AAA server (network element).
- A collection of authentication and dynamic authorization routing targets. Authentication requests are routed to a downstream AAA RADIUS server for authentication. Dynamic authorization requests are routed to the NAS device in the upstream network element.
- A SIC Diameter server that translates dynamic authorization requests (Change of Authorization [COA] and Disconnect Message [DM] requests) into vendor-specific attributes so that they can be understood by the NAS. The SIC Diameter server communicates with the SRC Diameter server for NAS routing information.
- Components supporting SNMP, statistics, and event logging.

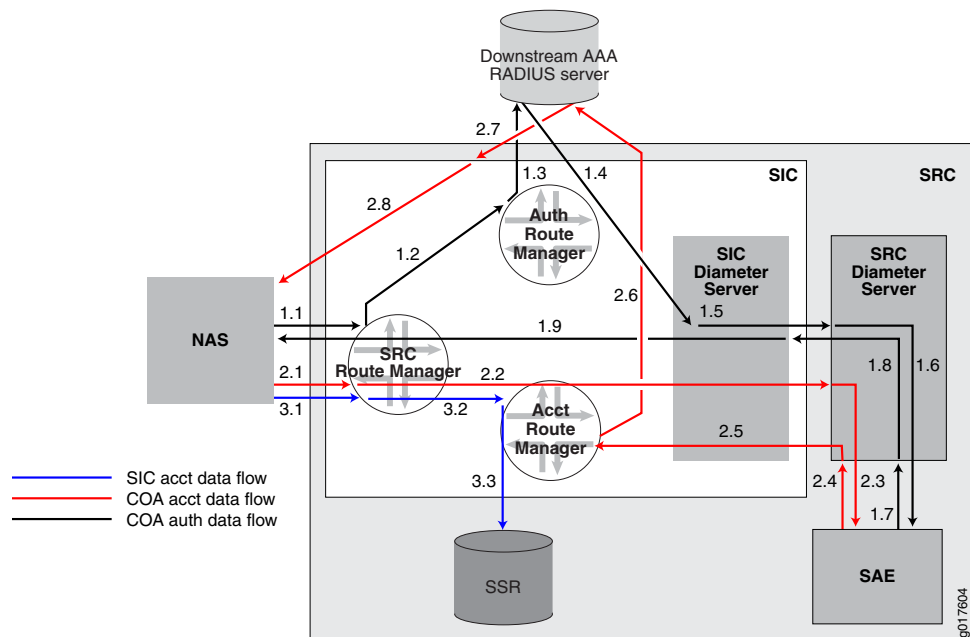
**Related
Documentation**

- [Accounting Method and Target \(SRC CLI\) on page 13](#)
- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 12](#)
- [Request Routing \(SRC CLI\) on page 14](#)

RADIUS Authentication/Authorization and Accounting Data Flow

Following is an overview of the SIC authentication, dynamic authorization, and accounting data flow processes. [Figure 1 on page 9](#) depicts the various functions involved in these processes.

Figure 1: Data Flow



The SIC internal functions include:

- **Authentication route manager**—The authentication route manager distributes RADIUS access requests to a downstream AAA RADIUS server (authentication target) based on the configured authentication routes. The SIC does not authenticate requests by itself. It proxies access requests to a downstream AAA RADIUS server. You can have multiple downstream AAA RADIUS servers in different realms. The SIC needs to send the access request to the correct RADIUS server based on the configured authentication routes, which are usually based on realm information.
- **Accounting route manager**—The accounting route manager distributes accounting requests to accounting targets, which could be either a downstream AAA RADIUS server or the SSR, based on configured accounting routes. Similar to RADIUS authentication, there may be multiple RADIUS accounting servers in different realms. The SIC needs to forward accounting requests to the correct RADIUS server based on the configured accounting routes. The SSR is just another accounting target to which the accounting route manager can direct accounting requests.
- **SRC route manager**—Because accounting requests may be destined to the COA path or the SSR path, the SIC needs a route manager to distribute accounting traffic to the two paths based on some routing information. The route manager receives routing information from the SRC Diameter server after the Diameter connection is established with the SIC. The routing information is configured in the SRC CLI under **[shared network nas-group name routes]**. When the SIC receives a RADIUS access or accounting request, the request is sent to the SRC route manager, which matches the request against each route received from SRC Diameter servers. If a route is matched, the request is sent to the COA path. If the request is an access request but no route is matched, the SIC still sends the request to the downstream AAA servers. However, the access response from

the AAA servers is returned to the NAS. If the request is an accounting request and no route is matched, the request is distributed to the SSR path.

COA Authentication Data Flow

The SIC needs to be in the RADIUS authentication path to insert the RADIUS class attribute in the Access-Accept response. The class attribute contains the encoded Diameter session ID as well as other information. The Diameter session ID is used to correlate service accounting requests to the SAE user session.

The numbers in the following procedure correlate to the numbers in [Figure 1 on page 9](#).

- 1.1 A RADIUS access request is received by the SIC.
- 1.2 The SRC route manager locates the responsible SRC Diameter server by using the routes configured under **[shared network nas-group name routes]**. Regardless of whether the SRC route manager finds a matching route, the SIC always sends the authentication request to a downstream AAA RADIUS server. The request is sent to the SIC authentication route manager to find the correct downstream AAA RADIUS server responsible for the request.
- 1.3 The authentication route manager locates the downstream AAA RADIUS server by using configured authentication routes and sends the request to the RADIUS server for authentication.
- 1.4 The SIC receives the authentication response from the downstream AAA RADIUS server. If no matching route is found in step 1.2 or the response is Access-Reject, the SIC sends the response to the NAS.
- 1.5 If an Access-Accept message is received from the downstream RADIUS server, the SIC sends an AA-Request (AAR) to the SRC Diameter server (through the SIC Diameter server), which owns the route matching the request.
- 1.6 The SRC Diameter server forwards the AAR to SAE by using CORBA.
- 1.7 The SAE creates the user session based on the AAR, activates activate-on-login (AOL) services for the user session, and returns AA-Answer (AAA) to the SRC Diameter server. The AAA message contains the service template name and arguments for the AOL services.
- 1.8 The SRC Diameter server sends the AA-Answer message to the SIC in a Diameter message.
- 1.9 The SIC Diameter server translates the service activation requests in the Diameter AA-Answer message to RADIUS attributes based on the configured device model. The SIC sends an Access-Accept RADIUS response to the NAS with the class attribute that contains the encoded Diameter session ID. Depending on the configuration, there may be multiple rounds between the SIC and SAE to exchange service activation information before the SIC sends the Access-Accept response to NAS.

COA Accounting Data Flow

After a subscriber is authenticated, the NAS sends an Accounting-Request (ACR) message for the user session and for every service that is activated. The accounting requests must contain the class attribute returned with the Access-Accept response.

The numbers in the following procedure correlate to the numbers in [Figure 1 on page 9](#).

- 2.1 The SIC receives an ACR message from the NAS.
- 2.2 The SRC route manager locates the responsible SRC Diameter server by using the routes configured under **[shared network nas-group name routes]**. When the SRC route manager locates a match, it sends the request as an ACR message to the SRC Diameter server (through the SIC Diameter server) corresponding to the route. If the SRC route manager does not find a match, the request is sent to the SSR path (see [“SIC Accounting Data Flow \(Accounting Target=Proxy\)” on page 11](#)).
- 2.3 The SRC Diameter server forwards the ACR message to the SAE in CORBA.
- 2.4 The SAE updates the user session with accounting information in the ACR message and sends an Accounting-Answer (ACA) message to the SRC Diameter server.
- 2.5 The SRC Diameter server forwards the ACA message to the SIC Diameter server.
- 2.6 The SIC receives the ACA message and needs to send the accounting request to the responsible downstream AAA RADIUS server. The SIC looks up the RADIUS server in the accounting route manager based on the configured accounting routes. The accounting route manager forwards the request to the downstream AAA RADIUS server. If accounting routes are not properly configured, the accounting route manager can forward the accounting request to the SSR. This is typically not desirable.
- 2.7 The downstream AAA RADIUS server sends an accounting response to the SIC.
- 2.8 The SIC sends the accounting response to the NAS.

SIC Accounting Data Flow (Accounting Target=Proxy)

- 3.1 The SIC receives an accounting request from the NAS.
- 3.2 The SRC route manager cannot locate a match (either because no SRC Diameter server is connected or because the connected SRC Diameter servers do not have any route matching the request), so it sends the accounting request to the accounting route manager.
- 3.3 The accounting route manager sends the request to a downstream AAA RADIUS server which can be another SIC.



NOTE: If SRC routes configured under **[shared network nas-group name routes]** for the SRC nas-group, and the SIC accounting routes are configured properly, the SIC can process accounting requests by using either a downstream AAA server.

- Related Documentation**
- [Configuring AAA Policies \(SRC CLI\) on page 152](#)
 - [Configuring the SAE to Manage AAA Devices on page 150](#)
 - [Configuring the NAS Groups \(SRC CLI\) on page 145](#)
 - [How the Dynamic Authorization Process Works in the SIC on page 38](#)

Local and Shared Configurations for the SIC (SRC CLI)

For the SIC, you need to define both a local and a shared group configuration.

Local Configuration

A local configuration applies to a specific server instance in the SIC group. The local configuration specifies the name of the server and the properties the server uses to connect to the Juniper Networks database where the configuration is stored. You specify the local server name by using the **edit slot number sic server** statement. You specify the connection properties for the Juniper Networks database by using the **slot number sic initial directory-connection** statement.

Shared Configuration

The SIC shared group configuration contains the configuration used by a group of servers. Each SIC server must belong to a group. The SIC group configuration controls the properties for the accounting methods, authentication route targets, dictionaries, editing rules, and RADIUS and Diameter options.

You create the SIC shared group configuration by using the **slot 0 sic server name /group-name/server-name** statement. The identifier associated with the group is the name of the shared configuration. This statement creates the shared group configuration and populates the server configuration with default data. Use this command to add servers to the group and populate the server with default data.

In addition, certain configuration options applicable to the individual server instances belonging to the group are also stored in the shared group configuration under the individual server name. These configuration options include the accounting and authentication routing rules, the event logging configuration, and the RADIUS and Diameter transport configurations specific to the server instance. You configure these options by using the **edit shared sic group identifier server** statements.

For example, if you want to create an SIC group named **server-group1** that includes a server named **server-bldg5**, from configuration mode:

- Specify *group-name* and *server-name*.

```
[edit]  
user@host# edit slot 0 sic server  
set name /server-group1/server-bldg5
```

The following rules depict how a new SIC group or server configuration is created on successfully committing the configuration:

- If the **group-name** does not exist in the Juniper Networks database, a new group and server instance as specified in this statement are created and populated with default data.
- If the **group-name** already exists in the Juniper Networks database, a server instance as specified in this statement is created under the group and populated with default data.

If you want to add another server to server-group1 named **server-bldg5a**, execute:

```
[edit]
user@host# edit slot 0 sic server
set name /server-group1/server-bldg5a
```

Creating a server by using this statement populates it with default data. You can also add a new server to an existing group by using the **shared sic group identifier server identifier** statement. However, this statement does not populate the server with default data.

Related Documentation

- [Creating an SIC Server Instance \(SRC CLI\) on page 51](#)
- [Creating an SIC Group and Server \(SRC CLI\) on page 50](#)

Accounting Method and Target (SRC CLI)

The proxy accounting method forwards accounting events to a downstream network element that contains a proxy AAA server.

You configure *accounting target* by specifying the accounting method used by the SIC group. The accounting target for explicit accounting routing rules and implicit routing rules is always a proxy AAA server in both downstream and remote network elements.

Using the Proxy RADIUS Accounting Method

The proxy RADIUS accounting method forwards accounting events to an accounting target (AAA server) located in a downstream network element. You use the **shared sic group identifier accounting-method accounting-method-name proxy radius** statement to configure a proxy RADIUS accounting method as the accounting method. You configure the downstream network element that contains the AAA server by using the **shared sic group identifier radius network-element id downstream** statement. You configure the AAA server as the accounting target by using the **shared sic group identifier radius network-element id downstream accounting-target** statement.

Following is a basic working configuration for the proxy accounting method that includes the default configuration:

- Accounting listener—You must specify a name for the accounting listener. You can use the default port 1813.
- Device model—You can use the default model.
- Proxy accounting method—You must configure the proxy accounting method.

- Outbound transport—You can use the default outbound transport.
- Upstream RADIUS network element and accounting client—You must define the upstream network element and at least one accounting client.
- Downstream RADIUS network element and accounting target—You must define the downstream network element and the accounting target.
- Accounting route—You must define the accounting route.
- Logger—You can use the default logger.

Related Documentation

- [Configuring Proxy RADIUS as the Accounting Method \(SRC CLI\) on page 82](#)
- [Request Routing \(SRC CLI\) on page 14](#)
- [Configuring Downstream Network Elements and Accounting and Authentication Targets \(SRC CLI\) on page 70](#)
- [Configuring the Optional Editing Rules Used by the SIC Group \(SRC CLI\) on page 80](#)

Authentication Route Targets (SRC CLI)

A downstream AAA server is responsible for authenticating all authentication requests. When the SIC receives a RADIUS authentication request, it evaluates authentication routes to determine which downstream AAA server is responsible for the request and then routes the request to the server. You configure the downstream AAA server as an authentication target by using the **shared sic group identifier radius network-element id downstream (authentication | accounting) authentication-target name** statement. You can configure multiple authentication targets. You can also define any number of authentication routes. The same routing conditions available for accounting routes can be configured for authentication messages.

Related Documentation

- [RADIUS Authentication/Authorization and Accounting Data Flow on page 8](#)
- [Accounting Method and Target \(SRC CLI\) on page 13](#)
- [Request Routing \(SRC CLI\) on page 14](#)

Request Routing (SRC CLI)

SIC routing rules define how the SIC routes accounting and authentication requests. You configure routing rules for each server in the SIC group. There are two types of routing rules:

- Explicit routing rules
- Implicit routing rules

Explicit Routing Rules

An explicit route is a collection of criteria used to select a particular routing target. Explicit routing rules consist of a condition, or set of conditions, and an accounting or authentication target to which the request is to be routed. Routing criteria consist of a list of simple Boolean expressions on RADIUS attributes and transactional variables. The accounting target is a downstream network element that contains an AAA server. The authentication target is an AAA server in a downstream network element.

You specify explicit routing rules based on the following match conditions:

- Realm name
- User identity
- Request attribute

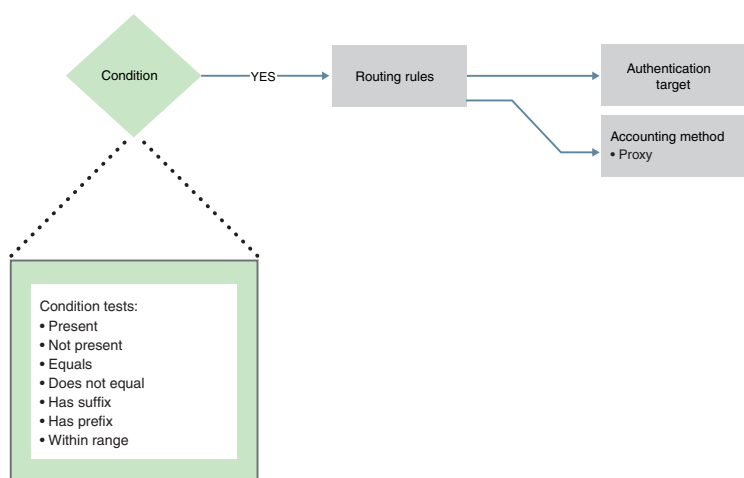
You can test the value of the match condition for the following conditions:

- Present
- Not present
- Equal
- Does not equal
- Has prefix
- Has suffix
- Range

The Has prefix and Has suffix condition tests work only on the string representation of the value. To test for a range condition, specify a low value and a high value.

[Figure 2 on page 15](#) depicts the explicit routing rule process.

Figure 2: Explicit Routing Rule Process



When the SIC receives an accounting or authentication request, it evaluates any defined explicit routing rules in the order they were configured. When multiple routes are configured, they are evaluated in the order they are displayed by the **show** command. A newly created route is displayed last among the routes and has the lowest priority, so it is evaluated last. You can use the SRC CLI **insert** command to move a route before or after another route to change its evaluation order. The higher a route is displayed on the list, the sooner it is evaluated. For a route to be selected, *all* conditions of the rule must be true. If a match is not found, the next configured rule is examined, and so on. As soon as a rule with matching criteria is encountered, the iteration stops and the accounting or authentication target in that rule is selected as the destination for the request. If a match for all conditions cannot be found in the explicit routing rules, the implicit routing rules are examined.

Before the request is sent to the specified target, you can edit it by optionally specifying an editing rule for the accounting route.

An example of explicit routing rule is:

If the NAS-Identifier is nas2 then the target is accounting method method2, which is the proxy accounting method, pointing to the RADIUS network element rne1:

```
[edit shared sic group group1 accounting-method method2 proxy]
user@host# show
radius {
  network-element rne1;
}
...
[edit shared sic group group1 server server1 accounting-route route2]
user@host# show
target method2;
condition {
  attribute nas-identifier;
  equals nas2;
}
```

This example is an accounting route example. Authentication routes work the same way.

Implicit Routing Rules

Implicit routes are realm based. You configure implicit routes by defining a network element that contains a remote AAA server and assigning it the proxy function. You can then either define a default route used for all requests from all realms or specify that only requests from specific realms are routed to the proxy AAA server. When you specify realms, you have the option to set a condition of either an exact match of the realm string, or a match on the prefix of the realm string.

Implicit routing rules have lower priority and are evaluated only if a match is not found for explicit routing rules. When a request is received, the SIC server evaluates the associated routing rules. First, the server evaluates any explicit routing rules. If no match is found, the server evaluates the implicit routing rules. When a match is found, the server processes the request by routing it to the specified network element that has the proxy function assigned to it.

Related Documentation

- [Configuring Explicit Routing \(SRC CLI\) on page 85](#)
- [Configuring Implicit Routing \(SRC CLI\) on page 88](#)
- [Configuring the Optional Editing Rules Used by the SIC Group \(SRC CLI\) on page 80](#)
- [SIC Editing Rules \(SRC CLI\) on page 17](#)
- [Accounting Method and Target \(SRC CLI\) on page 13](#)

SIC Editing Rules (SRC CLI)

Before the SIC sends the request to the specified accounting or authentication target, the request can optionally be edited according to the editing rules associated with the selected routing rule. Editing rules are similar to routing rules, in that the request is examined for matching conditions, and if one is found, the request is edited and then sent to the accounting target.

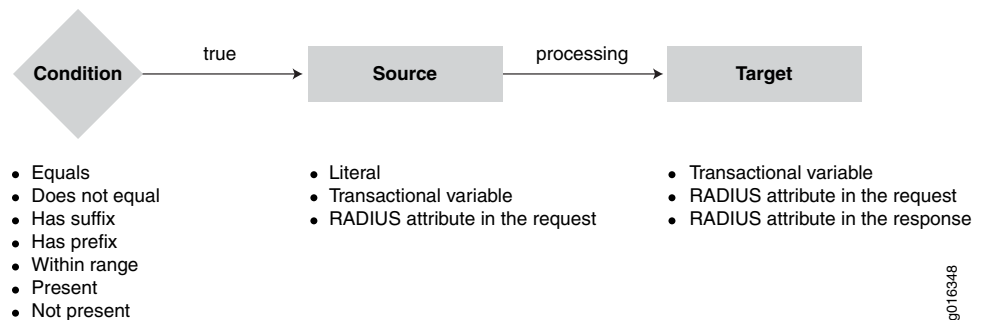
In addition to editing RADIUS attributes, the SIC can edit transactional variables and literals. Editing rules can define new transactional variables, in addition to certain built-in variables such as the result of username parsing, network access server (NAS) client lookup, and so on.



NOTE: You can control the number of transactional variables the SIC supports. The default value is 255. When you change this limit, you need to restart the SIC.

Figure 3 on page 17 depicts the SIC editing rule process.

Figure 3: SIC Editing Rule Process



You configure editing rules by defining the source and its associated match conditions, the editing conditions applied to the source value, and the target in which the edited result is placed. First, the SIC examines the specified source in the request for the defined match conditions. If all conditions are found to be true, the SIC edits the source value based on the defined editing conditions. The result is then placed in the defined target. The edited request, including both the original source and the new target value, is sent to the target.

Each editing rule is a simple assignment of a source (RValue) and a target (LValue). In any assignment the target can be one of the following:

- Transactional variable
- RADIUS attribute in the request
- RADIUS attribute in the response

The source can be one of the following:

- Literal
- Transactional variable
- RADIUS attribute in the request

The match conditions that you can test for in the source include whether a specific realm, user identity, or request attribute is:

- Present
- Not present
- Equals
- Does not equal
- Has suffix
- Has prefix
- Within range

If a match condition is found on the source, you can append or replace the value of the source and place it in the target.

Additionally, if the source is a request attribute, you can edit the value of the source by removing the suffix or prefix, or removing what is before or after the @ and place the result in the target. The remove before @ and remove after @ options can contain wildcards.

For example, if the request contains johnsmith@abcd.net:

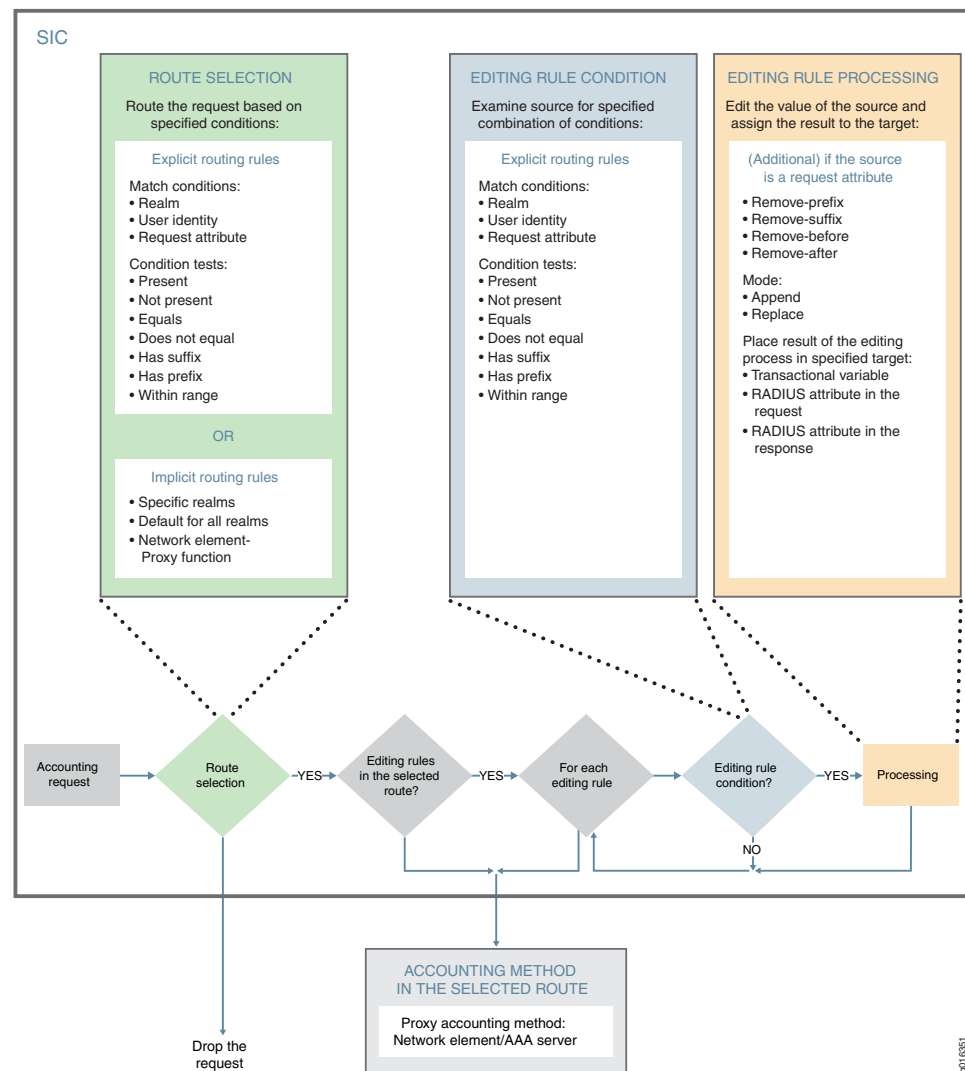
- Removing the prefix john results in: smith@abcd.net
- Removing the suffix .net results in: johnsmith@abcd
- Remove the attribute before the @ results in: abcd.net
- Remove the attribute after the @ results in: johnsmith

The SIC supports binary-level editing as follows:

- When a source attribute is of an octets type in the dictionary, the value of the attribute is encoded in hexadecimal format with a {hex} prefix. Operations including remove-before, remove-after, remove-prefix, and remove-suffix for such attributes work on their hexadecimal format. The search pattern used by the preceding operations must also use the hexadecimal format. For example, if the source string contains "@" and you want to remove everything after "@", the search pattern is 40, which is the ASCII number of @.
- A remove-before operation on an octets type attribute puts the {hex} prefix after the operation. A remove-prefix operation on an octets type attribute must include {hex} in the search pattern and the {hex} prefix is not added back after the operation. If you want to keep the {hex} prefix, use the remove-before operation.
- When converting an octets type attribute to another type, the SIC first tries to convert the attribute by using the binary representation of the target type. For example, the ip-address type attribute uses four octets. If the source attribute also uses four octets, the conversion is successful.
- If the binary conversion is not successful, the SIC tries the conversion using a printable string representation. For example, the printable format for the ip-address type attribute is "x.x.x.x". The conversion fails if the octets attribute value does not follow the printable format of the target attribute.

Figure 4 on page 20 depicts the editing rule process and the accounting route selection process. Authentication requests can also use this editing process. The target for authentication requests is always a downstream network element that includes an AAA server target.

Figure 4: Editing and Accounting Routing Rule Conditions and Processes



Example of an editing rule:

- If Unisphere-Virtual-Router is present, then the transactional variable vpn-id is the substring after ":" in Unisphere-Virtual-Router.
- If NAS-Identifier is nas3, then the transactional variable vpn-id is the realm portion of User-Name (realm transactional variable). Otherwise, the transactional variable vpn-id is the NAS-Identifier.

```
[edit shared sic group group1 editing edit1]
user@host# show

target {
variable vpn-id;
}
source {
```

```

request-attribute Unisphere-Virtual-Router {
  condition {
    attribute Unisphere-Virtual-Router;
    check-presence;
  }
  remove-before *:
}
  variable realm {
    condition {
      attribute nas-identifier;
      equals nas1;
    }
  }
}
default {
  request-attribute nas-identifier ;
}

```

Related Documentation

- [Configuring the Optional Editing Rules Used by the SIC Group \(SRC CLI\) on page 80](#)
- [Configuring Explicit Routing \(SRC CLI\) on page 85](#)
- [Accounting Method and Target \(SRC CLI\) on page 13](#)
- [Configuring Proxy RADIUS as the Accounting Method \(SRC CLI\) on page 82](#)

RADIUS and Diameter Configuration for the SIC Overview (SRC CLI)

The RADIUS and Diameter configuration for the SIC group consists of:

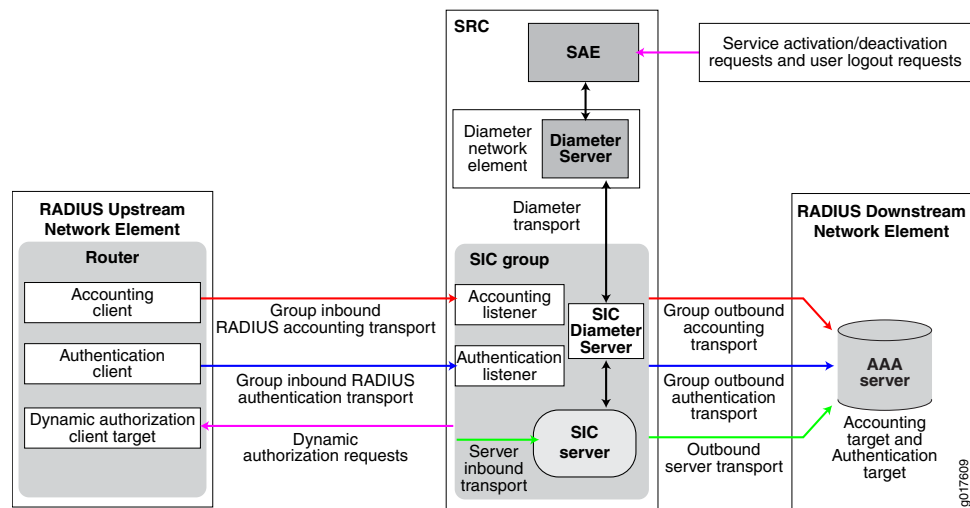
- RADIUS accounting and authentication listeners
- SIC Diameter server
- UDP transports for the group and each SIC server
- At least one RADIUS upstream network element with an accounting client, authentication client, and a dynamic authorization target
- At least one RADIUS downstream network element with an accounting target and an authentication target
- (Optional) A proxy function—Used for defining implicit routing rules



NOTE: Authentication clients and targets and dynamic authorization targets are optional and required only if you are supporting COA and DM requests.

Figure 5 on page 22 depicts the SIC RADIUS and Diameter configurations, which are detailed in the following sections.

Figure 5: SIC RADIUS and Diameter Configuration



RADIUS Accounting and Authentication Listeners

The SIC includes accounting and authentication listeners that listen for RADIUS accounting and authentication messages from the NAS and filter undesired events based on attachment session attributes. You must configure at least one accounting listener for the SIC group. If you are supporting COA or DM requests, you must also configure at least one authentication listener. To configure the listeners, you specify the UDP port that the SIC listens on as well as other parameters that control the receipt of UDP packets. The configuration options associated with the listeners control the RADIUS inbound transport for the SIC group, which is used to communicate with upstream network elements that contain one or more accounting and authentication clients.

SIC Diameter Server

The SIC includes a Diameter server. The SIC Diameter server communicates with the SRC Diameter server, which is a peer to the SIC Diameter server. The SIC Diameter server provides the translation between the SAE and SIC by translating COA or DM into VSAs so that they can be understood by the NAS. The SRC Diameter server also passes routing information from the SAE to the SIC Diameter server. This routing information is configured in the SRC CLI under `[shared network nas-group name routes]`.

Typically, the SIC connects to more than one SRC Diameter server. The Diameter servers may belong to a different redundant group. Each redundant group manages one or more NAS groups. Depending on the configuration, the SIC may connect to the SRC Diameter server in the same C Series Controller.

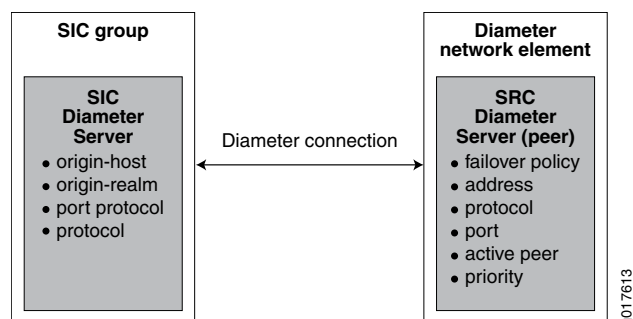


NOTE: Because both the SIC Diameter server and the SRC Diameter server need to listen to Diameter traffic, they should use different ports when both are active in the same C Series Controller.

SIC Diameter Server Configuration Overview

To configure the SIC Diameter server, you need to configure the server identity, port, and protocol. To configure the SRC Diameter server as a peer, configure the Diameter network element, the failover policy information, address, protocol, and active peer information. [Figure 6 on page 23](#) depicts the Diameter configuration for SIC.

Figure 6: SIC Diameter Configuration



RADIUS Network Elements

A network element is an addressable, logical network entity that contains RADIUS clients and targets.

An upstream RADIUS network element contains:

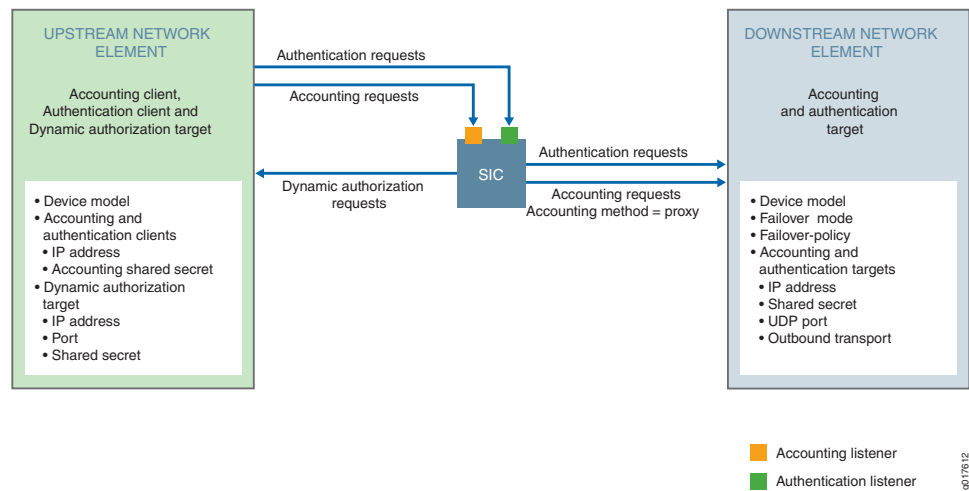
- Accounting clients, which send accounting messages to the SIC accounting listeners
- Authentication clients, which send authentication requests to the SIC authentication listeners
- Dynamic authorization targets, which receive COA/DM requests from the SIC

A downstream RADIUS network element contains an AAA server (target), which receives accounting and authentication messages from the SIC.

Network elements can contain multiple clients and targets.

[Figure 7 on page 24](#) depicts network elements and the various clients and targets they contain.

Figure 7: Upstream and Downstream Network Element Clients and Targets

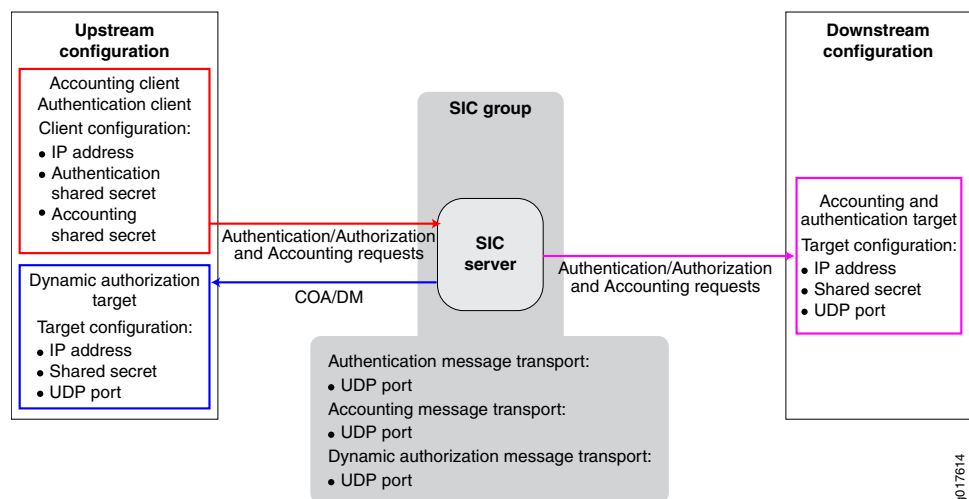


Configuring Upstream RADIUS Network Elements Overview

You need to configure at least one upstream network element containing at least one accounting client and one authentication client. If you are supporting dynamic authorization requests (Change of Authorization [COA] or Disconnect Messages [DMs]), you also need to configure a dynamic authorization target in the upstream network element. For dynamic authorization targets, you also need to configure the failover policy and mode. You configure upstream network elements by using the **shared sic group group-name radius network-element upstream** statement and specifying the shared secret, IP address, and device model of the accounting or authentication client.

Figure 8 on page 24 depicts RADIUS network element upstream and downstream client and target configuration options.

Figure 8: Upstream and Downstream Network Element Client and Target Configuration Options



Configuring Downstream RADIUS Network Elements Overview

You need to configure a downstream network element for the accounting and authentication targets. [Figure 8 on page 24](#) depicts RADIUS network element upstream and downstream configurations. You configure downstream network elements by using the **shared sic group *group-name* radius network-element downstream** statement and specifying the outbound transport, UDP port, shared secret, and IP address of the accounting target. You also need to specify the failover policy, failover mode, and the device model of the accounting target (AAA server).

Using the Proxy Function to Define Implicit Routing Rules

You use the proxy function to define implicit routing rules for accounting and authentication requests by specifying a remote AAA server as a proxy and having the SIC forward accounting and authentication requests to it. When the SIC receives a request, it first evaluates any explicit routing rules. If no match is found, it evaluates implicit routing rules. If a match is found, the SIC routes the request to the proxy AAA server.

You configure the proxy function by configuring a network element and specifying it as a proxy. You can then either define a default route used for all requests from all realms, or you can specify that only requests from certain realms are routed to the proxy AAA server. When you specify realms, you have the option to specify a match condition of either an exact match of the realm string or a match on the prefix of the realm string.

Related Documentation

- [Configuring the RADIUS Accounting Listener for the SIC Group \(SRC CLI\) on page 54](#)
- [Failover Policy on page 25](#)
- [Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\) on page 58](#)
- [Configuring Implicit Routing \(SRC CLI\) on page 88](#)
- [Request Routing \(SRC CLI\) on page 14](#)
- [RADIUS and Diameter Transports on page 27](#)

Failover Policy

The failover policy manages how the SIC sends messages over multiple paths to upstream and downstream network elements, and how it responds when it does not receive a response from a target in the network element within a specified amount of time. You configure a failover policy for the following targets:

- Accounting targets in downstream network elements
- Authentication targets in downstream network elements
- Dynamic authorization targets in upstream network elements

When multiple paths are configured to a network element, you need to specify the order in which the SIC uses the paths to send messages to the target. When the SIC sends a message to one of these targets, it expects to receive a reply within a certain amount of time as specified by the fast fail policy. If it does not receive the reply in the specified

time, it places the target into fast fail mode and rejects the request. You configure the failover policy by specifying the fast fail and retry parameters, which control such options as minimum number of times the server retransmits messages to the accounting, delay between sending retransmissions, as well as various timeout and delay settings that control how fast the server goes in and out of fast fail mode.

When the SIC has messages to send to a target, it first examines what failover mode is configured—either round-robin or primary or backup. It then examines whether all paths to the target are operational. It then sends the messages accordingly (over whatever paths are operating). As such, these features inherently manage communication failures by adjusting what paths are used when one or more paths are not working.

Failover Mode

Failover mode manages how the SIC sends messages over multiple paths to a network element target. You can configure failover mode for either round-robin or primary or backup. When the server has a message to send, it first examines whether failover mode is set for round-robin or primary or backup. Next, it examines whether all paths to the network element where the target resides are operational. It then sends the message over whatever path is operational based on failover mode.

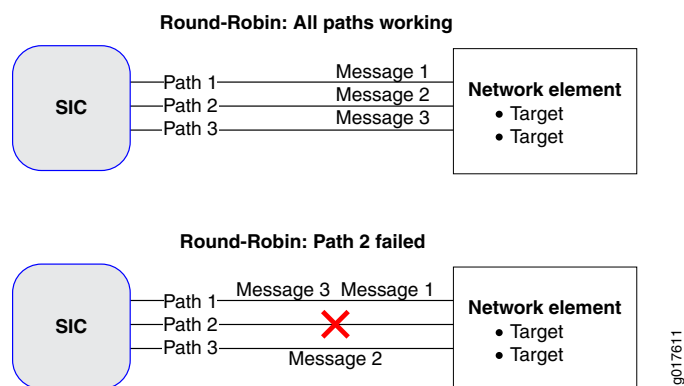
Round-Robin

When failover mode is set to round-robin, the SIC alternates the path it uses to send messages to the target.

Figure 9 on page 26 illustrates how the Round-Robin feature operates when all paths are working properly (top portion), and how it operates when one of the paths has failed (bottom portion).

With all three paths operating properly, if the SIC received three messages, the first message would be sent over path 1, the second message would be sent over path 2, and the third message would be sent over path 3. The next message received would be sent over path 1, and so on. However, if the server received three messages and path 2 had failed, the first message would be sent over path 1, the second message would be sent over path 3, and the third message would be sent over path 1.

Figure 9: Round-Robin



Primary or Backup

When failover mode is set to primary or backup, the SIC sends all messages over the first path defined in the ordered list. If the first path fails, all messages are sent over the next path in the ordered list. When the first path becomes operational again, all messages are again sent over it.

Related Documentation

- [Authentication Route Targets \(SRC CLI\) on page 14](#)
- [Dynamic Authorization Targets \(SRC CLI\) on page 42](#)
- [Configuring Failover Mode and Policy \(SRC CLI\) on page 73](#)

RADIUS and Diameter Transports

To support RADIUS accounting and authentication, you need to configure both inbound and outbound transports for the SIC group and each SIC server within the group.

To support dynamic authorization requests, you need to configure the RADIUS dynamic authorization transport as well as the Diameter transport to the Diameter server in SRC.

Inbound and Outbound RADIUS Accounting Transports for the SIC Group

To support RADIUS accounting in the SIC group, you need to configure the inbound transport by configuring the accounting listeners. You need to configure the group RADIUS outbound accounting transport if you are using the proxy accounting method, or if you are using implicit routing to forward accounting messages to a remote AAA server accounting target.

Inbound and Outbound RADIUS Authentication Transports for the SIC Group

To support RADIUS authentication in the SIC group, you need to configure the inbound transport by configuring the authentication listeners. You also need to configure the outbound RADIUS group authentication transport to the downstream AAA server authentication target.

SIC Server Inbound and Outbound RADIUS Transports

You can configure the inbound and outbound transport for each server in the SIC group. To do this, you specify the names you configured for the group inbound transport (authentication and accounting clients) and the group outbound transport and then specify the IP address the server used to send and receive UDP messages. This is optional. If it is not configured, the address for inbound transport is all IP addresses configured on the C Series Controller. If the address for the outbound transport is not configured, the SIC allows the SRC software to choose one from the list of addresses configured on the local machine.

Diameter Transport

You must configure bidirectional transport (Diameter connection) between an SIC server and the SRC Diameter server.

For the SIC server, you specify the origin-host, origin-realm, transport protocol (TCP or SCTP), and the port the SIC server uses for Diameter messages.

The SRC Diameter server must be defined within a network element. You need to specify the network element ID and the associated failover policy, as well as peer information for the SIC server including IP address, protocol (TCP or SCTP), and port. In addition, you must specify whether or not the peer is an active peer and the priority. These options are used when you have multiple connections from the Diameter server to multiple SIC servers (peers).

Related Documentation

- [RADIUS and Diameter Configuration for the SIC Overview \(SRC CLI\) on page 21](#)
- [Configuring the RADIUS Accounting Listener for the SIC Group \(SRC CLI\) on page 54](#)
- [Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\) on page 58](#)
- [Configuring the RADIUS Transport for an SIC Server \(SRC CLI\) on page 59](#)

SIC Dictionaries and Device Models Overview (SRC CLI)

The SIC uses dictionaries to define RADIUS attributes. Dictionaries identify the attributes the SIC expects when receiving RADIUS requests from a specific type of device—for example, an upstream NAS or downstream AAA server, and the attributes the SIC includes when sending a RADIUS response to a specific type of device. The SIC uses these definitions to parse accounting requests and generate responses.

Dictionaries and the Device Models Supported by the SIC Group

Each SIC group configuration must include a dictionary and a list of device models. When you configure the device model, you specify an identifier and the associated dictionary that the SIC uses when communicating with the device. The dictionary assigned to the device model identifies the attributes the SIC expects when receiving RADIUS requests from the specific device, and the attributes the SIC needs to include in responses to the device. The SIC uses these definitions to parse accounting, authentication, and dynamic authorization requests and generate responses.

In addition, when you configure an upstream or downstream network element, you need to specify which device models it supports based on the list of device models you have configured for the SIC group. Thereafter, whenever the SIC receives a RADIUS packet from the network element, it consults the associated dictionary for the attributes that it encounters in the packet.

Configuring Device Models and Their Associated Dictionaries for the SIC Group Overview

You need to specify the device models and their associated dictionaries for the SIC group and each network element the SIC needs to communicate with. To specify these for the SIC group, use the **shared sic group identifier model id** statement, and to specify these for network elements, use the **shared sic group identifier radius network-element id upstream** and the **shared sic group identifier radius network-element id downstream** statements.

Modifying a Dictionary

You can add attributes or modify existing attributes in dictionaries. However, you cannot delete the dictionary itself or any of the existing attributes. If you modify a dictionary, you need to restart the SIC for the change to take effect. Use the **shared sic group group name dictionary id** configuration statement to modify an existing dictionary.

Configuring the Dictionaries Used by the SIC Group Overview

The SIC includes standard RADIUS devices. All dictionaries implicitly import RADIUS standard attributes. The RADIUS dictionary is the default dictionary. It is loaded by default when you configure an SIC group and is sufficient for most environments.

- Related Documentation**
- [Configuring the Device Models Supported by the SIC Group \(SRC CLI\) on page 54](#)
 - [Configuring Dictionaries for the SIC Group \(SRC CLI\) on page 51](#)

SIC Local Realms Overview

Defining a realm as local to the SIC group instructs the SIC to use a local server to process the request. The network access identifier (NAI) in the request identifies the intended realm. To properly interpret requests received from intermediate servers, the SIC server must know which realms it is responsible for servicing locally.

When a request is received, the SIC examines the NAI to determine the realm to which the request is to be routed. If the realm is configured as a local realm to the SIC server, the request is processed by the local server. If no realm is present in the NAI, the request is considered to be local.

- Related Documentation**
- [Configuring What Realms Are Local to the SIC Group \(SRC CLI\) on page 75](#)
 - [Local and Shared Configurations for the SIC \(SRC CLI\) on page 12](#)
 - [Creating an SIC Server Instance \(SRC CLI\) on page 51](#)

SIC Event Logging Overview (SRC CLI)

SIC log streams capture different groups of server-related events at various levels of granularity. You can configure the SIC to capture any number of log streams. If you configure multiple log streams, make sure you configure unique names for each log stream by using the **shared sic group identifier server identifier logger identifier** statement. Each log stream you create captures events in a separate log file, which is date stamped, and you can also assign a prefix to it for easy identification.

Log messages are divided into several log groups according to the subject of the log information. You can configure a log stream to display only log messages from particular log groups. Each log group captures different types of server-related events. You configure

the level of granularity captured for the log group by setting the event level for the log group.

Log File Options

You use the configuration options described in [Table 5 on page 30](#) to define the properties of the log files.

Table 5: SIC Log File Options

Option	Description
filename	Prefix added to the log file name. This string is prepended to each log file name.
filter	Filter to define which event messages are logged or ignored. The filter specifies the logging level, such as debug. <ul style="list-style-type: none"> Error events are captured for every log group Debug events are captured for every log group
flush-after-writes	If set to true, log messages are immediately written to the log file without buffering. Use this setting for real-time logging. If set to false, SIC log messages are kept in the buffer until the buffer is full and then all messages in the buffer are written to the log file. Use this setting for performance optimization, when real-time logging is not needed.
footer	Footer message added to the end of each log file.
header	Header message added to the beginning of each log file.
high-resolution-timestamps	High-resolution time reporting system functions are used.
maximum-file-size	New log file created after these many bytes. When a log file reaches this size, logging begins in a new log file.
prepend-message-header	Prepend each log message with additional information. Add time, thread, and transaction information to each log message. You can achieve additional fine-tuning by using the work-id-label, work-id-padding, and utc options.
rollover-interval	New log file is created after this amount of time elapses. Specified in seconds.
rollover-on-startup	New log file is created every time the server starts.
utc	Time and date values reflect Universal Time Coordinates (UTC), formerly known as Greenwich Mean Time or (GMT). Otherwise, values reflect local time.
work-id-label	Work data ID prefix added to each log message.
work-id-padding	String added to each log message if work data is not available.

Event Levels

The event level specifies the level of detail captured for the log group. You configure the event level by specifying the log group and then specifying the associated event level. The event level you specify is the highest event level displayed for the log group. You can configure the log stream to display log items from levels at and below a particular event level.

Be careful when using event logging because it consumes server resources while capturing events, and consumes disk space to store the log files. We recommend that event logging be used primarily for troubleshooting purposes, and that you limit the amount of information captured in a log stream to control the consumption of server resources and disk space. Limiting the amount of information in the log stream also makes it easier to interpret the information in the log files. For example, you might configure one log stream to capture only configuration-related events by setting the Configuration log group event level to Detail, and setting all other log group event levels to Error.

In general, each event level includes less verbose event types. For example, if you configure an event level of Warning, then warnings and errors are logged to the specified log stream. The event levels in order of increasing verbosity are shown in [Table 6 on page 31](#).

Table 6: SIC Event Levels

Event Level	Description
None	No events will be logged for the log group.
Error	An error as an event that may cause the system to operate incorrectly. Examples include exceptions being thrown, an inability to continue processing a transaction, or configuration errors that cause a component to fail to start.
Warning	Errors and warnings are logged. Warnings are less severe but more verbose than errors, in that a warning should be logged when the system was able to handle an unexpected input or condition without any threat to the operation of the server. Examples of warnings include invalid packet contents or failures in contacting remote servers.
Standard	Errors, warnings, and standard messages are logged. Standard logging messages show events as a result of normal operation.
Detail	Messages in the log are shown at event levels error, warning, standard, and detail. Detail logging is intended to inform why and how the particular result indicated by standard logging was reached. Server components that perform significant processing on the transaction, such as determining validity of the packet contents, log details about decisions they made. All server components that route the transaction through different processing based on the nature of the transaction log their routing activity at this level. The detail log is allowed to refer to the contents of messages logged at the standard level; that is, it will never be read without the standard messages.
Debug	Messages in the log are shown at event levels error, warning, standard, detail, and debug. Debug logging is provided for engineering troubleshooting only.

Log Groups

Log groups specify the type of server functionality for which you want to log events. The log groups listed in [Table 7 on page 32](#) are available.

Table 7: SIC Log Groups

Log Group	Description
Administration	Reports events related to server administration, such as: <ul style="list-style-type: none"> • A server access log, including identity of the administrator. This is available using the standard event level. • Changes made to the server configuration, including identity of the administrator. This is available using the Detail event level.
Configuration	Reports events related to configuration.
Packet	Reports events related to transaction processing.
PacketTrace	Displays content of a packet in an <attribute name>:<attribute value> format.
PacketTraceRaw	Displays content of a packet in its raw (octets) format.
System	Reports events related to the system, such as: <ul style="list-style-type: none"> • Resource failures (no memory, file not found, disk full, and so on.) • Unknown exceptions • System start • System stop

Related Documentation

- [Configuring Event Logging for an SIC Server \(SRC CLI\) on page 88](#)
- [Configuring SNMP for the SIC Group \(SRC CLI\) on page 92](#)
- [SNMP Support for the SIC Overview \(SRC CLI\) on page 32](#)

SNMP Support for the SIC Overview (SRC CLI)

The Simple Network Management Protocol (SNMP) implementation for the SIC supports alerts and is based on an external SNMP agent that accesses the SIC server agent.

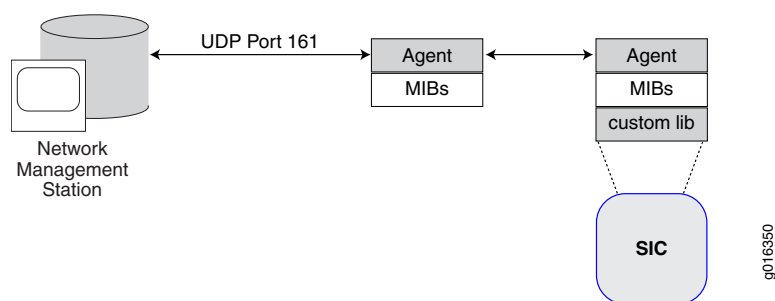


NOTE: You can query status using an SNMP Get command, and you can receive alert notifications through SNMP traps from the agent manager. However, you cannot configure the SIC server using SNMP because the SIC does not support the SNMP Set command.

SNMP support for the SIC functions in the same manner as other SRC components. The SIC has its own subagent, which communicates with the main SRC SNMP daemon using the AgentX protocol. The subagent registers with the main daemon for all relevant object identifiers (OIDs). Traps are communicated from the subagent to the main daemon.

[Figure 10 on page 33](#) depicts the SIC SNMP support.

Figure 10: SNMP Support for the SIC



Use the **show** command to view all statistics counters available to the SNMP daemon. [Table 8 on page 33](#) list the MIBs used by the SIC to maintain accounting statistics.

Table 8: MIBs Used by the SIC for SNMP Statistics

MIB	Description
RFC4670.mib	Maintains accounting client statistics.
RFC4671.mib	Maintains accounting server statistics.

The SIC supports the traps described in [Table 9 on page 33](#).

Table 9: SNMP Traps Supported for the SIC

SNMP Trap	Description
diameter-base-protocol-error	Diameter base protocol error occurred.
diameter-peer-connection-down	Diameter peer connection is down.
diameter-permanent-failure	Diameter permanent failure occurred.
diameter-transient-failure	Diameter transient failure occurred.
sic-server-internal-error	An SIC server implementation—dependent error occurred.
sic-server-log-file-failure	Operation on an SIC server log file such as opening, reading, or writing failed. This is most likely due to the server disk being out of space.
sic-server-resource-failure	An SIC server implementation—dependent resource failure occurred.

Table 9: SNMP Traps Supported for the SIC (*continued*)

SNMP Trap	Description
sic-server-shutdown	The SIC server process stopped successfully.
sic-server-startup	The SIC server process started successfully.
sic-server-unauthorized-administration-request	HTTP/HTTPs requests sent to the SIC server were denied because the user does not have proper permission to access the URL.

**Related
Documentation**

- [Configuring SNMP for the SIC Group \(SRC CLI\) on page 92](#)
- [Configuring Event Logging for an SIC Server \(SRC CLI\) on page 88](#)
- [SIC Event Logging Overview \(SRC CLI\) on page 29](#)

CHAPTER 3

RADIUS COA

- [Managing Dynamic Services on RADIUS-Enabled Devices on page 35](#)
- [SIC Dynamic Authorization Support Overview on page 36](#)
- [How the Dynamic Authorization Process Works in the SIC on page 38](#)
- [Dynamic Authorization Targets \(SRC CLI\) on page 42](#)

Managing Dynamic Services on RADIUS-Enabled Devices

When you integrate the SIC, you can manage services on RADIUS-enabled devices in an SRC network. The SIC processes messages between the NAS device and the RADIUS server. You can configure the services, policies, and parameters with the SRC software independent of the NAS device. The SRC Diameter server communicates with the SIC Diameter server by using Diameter messages to dynamically manage services (like COPS and JSRC) for a subscriber session. The SIC converts the RADIUS messages (such as Access-Request and Accounting-Request) to Diameter requests. The SIC Diameter server forwards the Diameter request (such as AA-Request and Accounting-Request) to SRC Diameter server, which then forwards the Diameter request to the AAA device driver. The SIC Diameter server converts the Diameter messages to RADIUS messages and routes dynamic RADIUS requests to the NAS device (client or target), or to the accounting or authentication target. For more information about the SIC authentication, dynamic authorization, and accounting data flow processes, see [“RADIUS Authentication/Authorization and Accounting Data Flow” on page 8](#).

The SIC Diameter server forwards messages to the SRC Diameter server, which then forwards them to the AAA device driver in the SAE. These Diameter messages perform the following functions:

- AAR—Attach the subscriber to the access network.
- ACR—Provide accounting information.
- ASR—Disconnect the subscriber.
- PPR—Start, modify, or stop the service session; send message routing configuration.
- STR—Detach the subscriber from the access network.

You must configure NAS groups and an AAA device driver for each NAS group hosted by the SAE. You also need to configure the services, policies, and parameters that the SIC

uses for service activation on the NAS device. You need to provide specific information for the service templates used by the SIC.

Service templates list the parameters needed for service activation on a NAS device. The SIC has detailed knowledge about the specific NAS device so that it can use the services, policies, and parameters configured by the SRC software for managing services on the NAS device.

Tasks to set up the management of services on RADIUS-enabled devices are:

- Configure the SIC. See [“SIC RADIUS Configuration Summary \(SRC CLI\)” on page 46](#).
- Configure the SRC Diameter application. See [“Configuring the Diameter Application \(SRC CLI\)” on page 137](#).
- Configure the NAS groups. See [“Configuring the NAS Groups \(SRC CLI\)” on page 145](#).
- Configure the SAE to manage AAA devices. See [“Configuring the SAE to Manage AAA Devices” on page 150](#).
- Configure AAA policies. See [“Configuring AAA Policies \(SRC CLI\)” on page 152](#).

**Related
Documentation**

- [SIC Dynamic Authorization Support Overview on page 36](#)
- [How the Dynamic Authorization Process Works in the SIC on page 38](#)
- [Subscriber Information Collector Overview on page 7](#)

SIC Dynamic Authorization Support Overview

The SIC can dynamically manage services on RADIUS-enabled devices. The RADIUS capabilities of the SIC allow the SRC software to be aware of the subscriber activity and make dynamic RADIUS requests using the following RADIUS features:

- Authentication, authorization, and accounting (AAA)
- Change of Authorization (COA) message
- Disconnect Message (DM)

The SIC uses RADIUS AAA messages to communicate with the RADIUS server and the network access server (NAS). The SIC converts Diameter messages to RADIUS messages and vice versa. The SIC also performs conversion between Diameter attribute-value pairs (AVPs) and RADIUS attributes.

The SIC can provide:

- Device abstraction and shared secrets for the NAS device
- Accounting and authentication support for subscriber sessions and service sessions
- COA and DM support
- Service parameter changes

RADIUS was designed as an AAA protocol in client/server mode. Supporting dynamic authorization requests requires that the SIC communicate Change of Authorization (COA) requests and Disconnect Messages (DM) to the network access server (NAS). However, every NAS vendor implements services by using different sets of vendor-specific attributes (VSAs); there is no universal language for sending requests to a NAS. To translate COA or DM requests into the correct dialect, the SIC uses service templates, which define services that the router activates and deactivates. These service templates translate COA or DM requests into VSAs so that the NAS device can understand and implement them. Service templates are created using the SRC CLI and they specify initial authorization, activation, deactivation, and abort session requests.

We provide device templates for Juniper Networks E Series Broadband Services Routers running JunosE Software release 7.2 or later and for Cisco routers running Cisco IOS Release 12.2SB. These templates include sample global and service templates that you can modify for your specific environment. If you want to add a router from another vendor, you must create a new template so that the SRC can communicate properly with your new router.

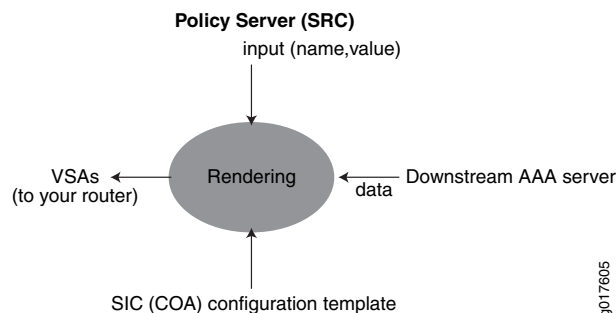
The SIC dynamic authorization function includes:

- RADIUS listeners for authentication and accounting requests.
- RADIUS dynamic authorization interface for sending COA or DM requests to the NAS.
- RADIUS proxy function for forwarding RADIUS authentication and accounting requests to a downstream RADIUS server.
- SIC Diameter server interacts with the SRC Diameter server. User access, accounting requests, and service accounting information are sent to the SAE through this Diameter interface.

Rendering

The SIC generates COA or DM requests on request from the SAE. Translations between SAE, SRC Diameter server, SIC, and your router must take place. This translation process is called rendering. The rendering process is shown in [Figure 11 on page 37](#).

Figure 11: The Rendering Process



The rendering process takes three inputs and produces one output. Inputs are:

- The data the SAE sends (to and from the SRC Diameter server)

- SIC configuration (device and service) templates
- Data that returns with the authentication response from the downstream AAA server (available only for initial authorization process)

**Related
Documentation**

- [How the Dynamic Authorization Process Works in the SIC on page 38](#)
- [Managing Dynamic Services on RADIUS-Enabled Devices on page 35](#)

How the Dynamic Authorization Process Works in the SIC

This section describes the process of creating device and service templates for dynamic authorization. To understand how service templates interact with service requests, there are three main scenarios that you need to consider:

- Initial Authorization
- Activation and Deactivation
- Abort Session

Each of these has a service template associated with it.



NOTE: In the following discussion and illustrations, the NAS communicates with the SIC through the router.

Introduction

There are two common behaviors that trigger dynamic authorization requests:

- The SIC sends a request to the SAE notifying it about an event, such as authentication success.
- The SAE requests a service, such as activation, deactivation, or abort session.

In the former case, the SAE replies, and the SIC uses this reply as one of the inputs to the rendering process to generate VSAs. In any case, the SAE supplies data that the SIC uses as one of the inputs to the rendering process to generate VSAs. The SIC then sends the VSAs to the NAS so that it can activate or deactivate services.

In the process, requests may go not only from the NAS to the SIC, but also to the downstream AAA server, to the SAE, and, in the case of the initial authorization scenario, from the SIC to the downstream AAA server.

Initial Authorization

Initial authorization of services requires that your NAS support service activation in the Access-Accept message. This capability is called **Initial-Authorization** mode in the service template. This scenario begins when the NAS sends an authorization request to the SIC. The SIC in turn sends a RADIUS access request to the downstream AAA server that handles authorization requests.

If the downstream AAA server approves the request, it sends a RADIUS Access-Accept message to the SIC. Using the global service template configuration, the SIC formats the authorization request to the SAE. At this point, the SAE replies with service activation data used as input to the rendering process. This data contains the service name as specified in the service template along with the attribute values and parameters. For example, if the SAE requests `content_provider_tiered` service, the SIC renders data by using the corresponding mode, as shown in the following example service template configuration:

```
service-template content_provider-tiered {
  mode Initial-Authorization {
    attributes {
      item attr1 {
        parameterized-attribute {
          format
content_provider_tiered($(contentProviderAddress),$(contentProviderMask),
$(subscriberAddress),$(subscriberMask),
$(upstreamBandwidth),$(downstreamBandwidth));
          name Unisphere-Activate-Service;
        }
      }
      item attr2 {
        default-attribute {
          name Unisphere-Service-Stats;
          value 1;
        }
      }
    }
  }
}
```

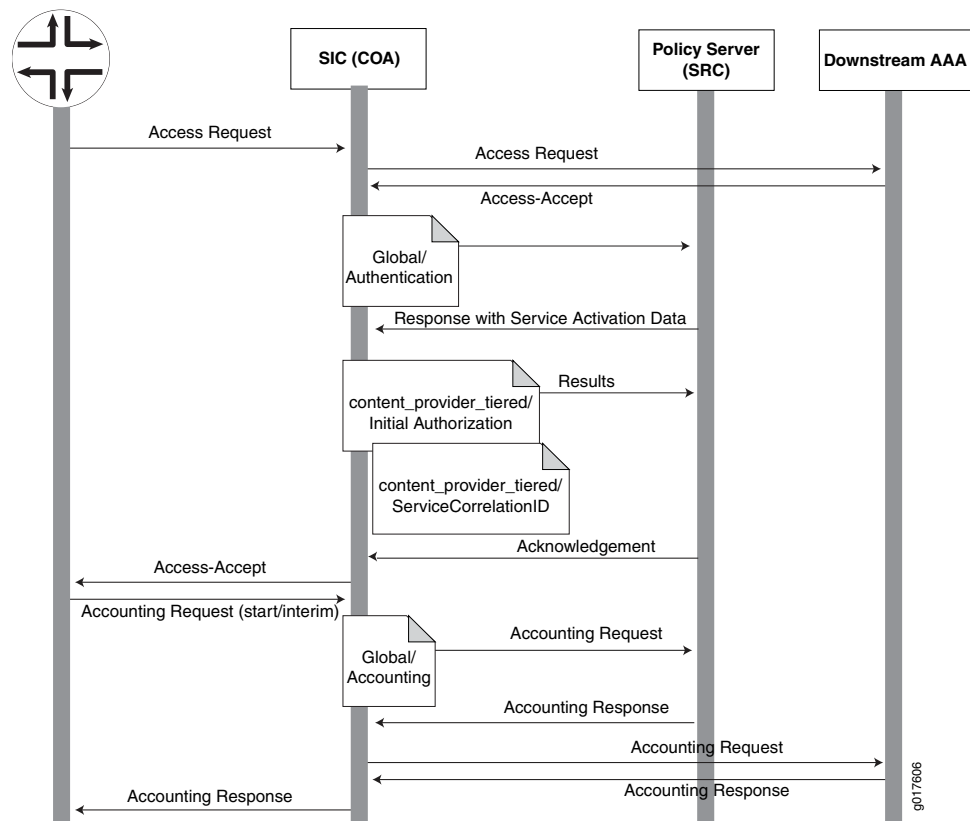
The SIC renders the Access-Accept message, then informs the SAE about rendering successes and failures in another request. The SAE sends an acknowledgement back to the SIC, which in turn sends a rendered Access-Accept message to the NAS.

Accounting

As soon as the requested service is active, the next step is sending an accounting (start or interim) request from the NAS to the SIC. Using the rendering process and the information defined in accounting mode in the global service template, the SIC sends an accounting request to the SAE, which then sends an accounting response. After receiving this response, the SIC sends an accounting request to the downstream AAA server, which sends an accounting response. Finally, the SIC sends an accounting response back to the NAS and service accounting is complete.

[Figure 12 on page 40](#) shows the initial authorization and accounting timing sequence. The rectangles with a folded corner represent pieces of the service or global service templates. For purposes of this illustration, the SIC and SRC are shown in two distinct rectangles.

Figure 12: Initial Authorization and Accounting Timing Sequence



Service Activation and Deactivation

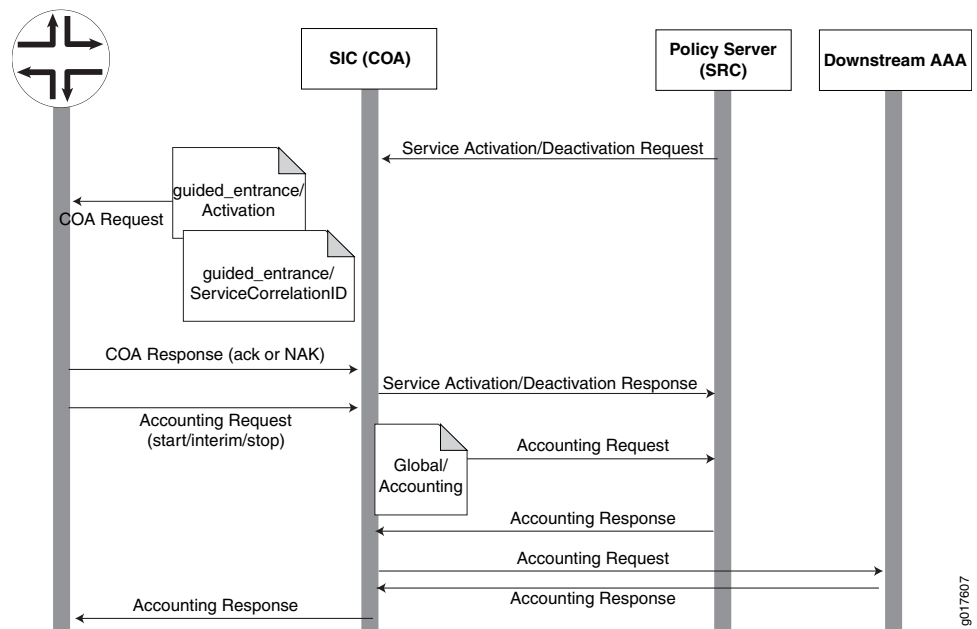
This section describes the service activation and deactivation scenarios. The sequences for activation and deactivation are identical except that the activation sequence uses activation requests and the deactivation sequence uses deactivation requests.

A service activation begins with an activation request from the SAE to the SIC. [Figure 13 on page 41](#) uses the `guided_entrance` service activation request as an example. This activation request includes all the information needed for the SIC to render the `guided_entrance` service activation request into RADIUS format for the NAS. The SIC sends the rendered request, along with a service correlation ID, as a COA to the NAS. The NAS responds with an acknowledgement packet (ack) or negative acknowledgement (NAK). The SIC then sends a service activation response to the SAE.

This completes the service activation. The NAS then initiates an accounting request, the timing sequence of which is identical to the sequence described in [“Accounting” on page 39](#).

[Figure 13 on page 41](#) shows the activation and deactivation timing sequences. For purposes of this illustration, the SIC and SRC are shown in two distinct rectangles.

Figure 13: Activation and Deactivation Timing Sequences



Abort Session Requests

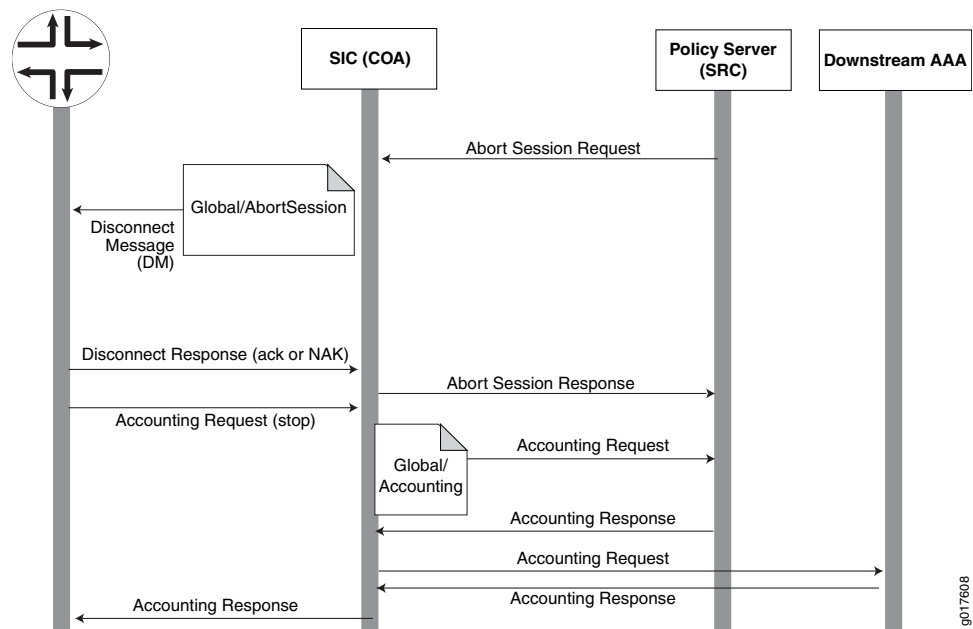
If the SAE receives an abort session request, it sends it to the SIC. The SIC, using the global service template and Abort-Session mode, renders the request and sends it, along with a service correlation ID, as a Disconnect Message (DM) to the NAS. The NAS responds with an ack or NAK. The SIC then sends a response to the SAE.

In all situations, abort session requests follow the same sequence and use the same global service template.

This completes the abort session scenario. The NAS then initiates an accounting request, the timing sequence of which is identical to the sequence described in [“Accounting” on page 39](#).

[Figure 14 on page 42](#) shows the abort session timing sequence. For purposes of this illustration, the SIC and SRC are shown in two distinct rectangles.

Figure 14: Abort Session Timing Sequence



- Related Documentation**
- [SIC Dynamic Authorization Support Overview on page 36](#)
 - [Device and Service Template Configuration Overview \(SRC CLI\) on page 95](#)
 - [Dynamic Authorization Targets \(SRC CLI\) on page 42](#)

Dynamic Authorization Targets (SRC CLI)

The NAS is considered a dynamic authorization target to the SIC. Dynamic authorization targets are configured in upstream network elements by using the **shared sic group identifier radius network-element id upstream dynamic-authorization-target** statement. When the SIC receives a COA or DM request, it processes the request based on the device and service and global service templates specified in the request.

- Related Documentation**
- [Device and Service Template Configuration Overview \(SRC CLI\) on page 95](#)
 - [SIC Dynamic Authorization Support Overview on page 36](#)
 - [How the Dynamic Authorization Process Works in the SIC on page 38](#)

PART 2

Configuration

- [Configuration Tasks for Subscriber Information Collector on page 45](#)
- [Configuration Tasks for RADIUS COA on page 95](#)
- [Example for Subscriber Information Collector on page 155](#)

CHAPTER 4

Configuration Tasks for Subscriber Information Collector

- [SIC Configuration Summary on page 45](#)
- [Configuring the Connection Between the SIC and the Juniper Networks Database \(SRC CLI\) on page 48](#)
- [Creating an SIC Group and Server \(SRC CLI\) on page 50](#)
- [Creating an SIC Server Instance \(SRC CLI\) on page 51](#)
- [Configuring Dictionaries for the SIC Group \(SRC CLI\) on page 51](#)
- [Configuring the Device Models Supported by the SIC Group \(SRC CLI\) on page 54](#)
- [Configuring the RADIUS Accounting Listener for the SIC Group \(SRC CLI\) on page 54](#)
- [Configuring the RADIUS Authentication Listener for the SIC Group \(SRC CLI\) on page 56](#)
- [Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\) on page 58](#)
- [Configuring the RADIUS Transport for an SIC Server \(SRC CLI\) on page 59](#)
- [Configuring the SIC Diameter Server \(SRC CLI\) on page 60](#)
- [Configuring Upstream and Downstream RADIUS Network Elements \(SRC CLI\) on page 64](#)
- [Configuring What Realms Are Local to the SIC Group \(SRC CLI\) on page 75](#)
- [Configuration Statements for SIC Editing Rules \(SRC CLI\) on page 76](#)
- [Configuring the Optional Editing Rules Used by the SIC Group \(SRC CLI\) on page 80](#)
- [Configuring the Accounting Method Used by the SIC Group \(SRC CLI\) on page 82](#)
- [Configuring the Authentication Target Used by the SIC Server \(SRC CLI\) on page 82](#)
- [Configuring Request Routing \(SRC CLI\) on page 83](#)
- [Configuring Event Logging for an SIC Server \(SRC CLI\) on page 88](#)
- [Configuring SNMP for the SIC Group \(SRC CLI\) on page 92](#)

SIC Configuration Summary

- [SIC RADIUS Configuration Summary \(SRC CLI\) on page 46](#)
- [SIC RADIUS Dynamic Authorization Configuration Summary \(SRC CLI\) on page 47](#)

- [SIC Diameter Configuration Summary \(SRC CLI\)](#) on page 48
- [Configuring Management of RADIUS-Enabled Devices for the SIC \(SRC CLI\)](#) on page 48

SIC RADIUS Configuration Summary (SRC CLI)

The SIC default configuration satisfies the needs of most environments, with minor changes such as accounting and authentication targets, editing rules, routing rules, and RADIUS clients.

To configure the SIC RADIUS options:

1. Use the default settings for the connection between the SIC and the Juniper Networks database, or configure your own connection.

See [“Configuring the Connection Between the SIC and the Juniper Networks Database \(SRC CLI\)”](#) on page 48.

2. Use the default SIC group, or create an SIC group or server.

See [“Creating an SIC Group and Server \(SRC CLI\)”](#) on page 50.

3. Use the default dictionary for the SIC group, or define dictionary changes.

See [“Configuring Dictionaries for the SIC Group \(SRC CLI\)”](#) on page 51.

4. Use the default device model (default-model), or configure the device models used by upstream and downstream network elements.

See [“Configuring the Device Models Supported by the SIC Group \(SRC CLI\)”](#) on page 54.

5. (Optional) Configure editing rules for the SIC group.

See [“Configuring the Optional Editing Rules Used by the SIC Group \(SRC CLI\)”](#) on page 80.

6. Configure any realms you want processed by the local server.

See [“Configuring What Realms Are Local to the SIC Group \(SRC CLI\)”](#) on page 75.

7. Configure the accounting listener for the SIC group.

See [“Configuring the RADIUS Accounting Listener for the SIC Group \(SRC CLI\)”](#) on page 54.

8. Configure the authentication listener for the SIC group.

See [“Configuring the RADIUS Authentication Listener for the SIC Group \(SRC CLI\)”](#) on page 56.

9. Configure outbound transport for the SIC group.

See [“Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\)”](#) on page 58.

10. Configure the upstream and downstream network elements.

For the upstream network element, configure:

- Supported device models
- Accounting client

- Authentication client
- Dynamic authorization target and failover policy

For the downstream network element, configure:

- Supported device models
- Accounting target and failover policy and mode
- Authentication target and failover policy and mode

See [“Configuring Upstream and Downstream RADIUS Network Elements \(SRC CLI\)”](#) on page 64.

11. Configure the accounting method used by the SIC group.
See [“Configuring the Accounting Method Used by the SIC Group \(SRC CLI\)”](#) on page 82.
12. (Optional) Create additional server instances as desired.
See [“Creating an SIC Server Instance \(SRC CLI\)”](#) on page 51.
13. (Optional) Configure request routing.
See [“Configuring Request Routing \(SRC CLI\)”](#) on page 83.
14. Configure transport options for each SIC server in the group.
See [“Configuring the RADIUS Transport for an SIC Server \(SRC CLI\)”](#) on page 59.
15. Configure event logging.
See [“Configuring Event Logging for an SIC Server \(SRC CLI\)”](#) on page 88.
16. (Optional) Configure SNMP options.
See [“Configuring SNMP for the SIC Group \(SRC CLI\)”](#) on page 92.

SIC RADIUS Dynamic Authorization Configuration Summary (SRC CLI)

To configure RADIUS dynamic authorization support, you must configure the device and service templates:

- Review the device and service template configuration overview.
See [“Device and Service Template Configuration Overview \(SRC CLI\)”](#) on page 95.
- Configure the device template used by the SIC group.
See [“Configuring Device Templates \(SRC CLI\)”](#) on page 102.
- Configure the device capabilities.
See [“Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\)”](#) on page 103.
- Configure the service template.
See [“Configuring SIC Service Templates \(SRC CLI\)”](#) on page 105.
- Configure any tagged attributes for the service template.

See [“Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\)”](#) on page 114.

- Configure the global service template.

See [“Configuring Global Service Templates \(SRC CLI\)”](#) on page 122.

SIC Diameter Configuration Summary (SRC CLI)

To configure Diameter support for the SIC:

1. Configure the SIC Diameter server including the Diameter network element failover policy and the Diameter peers.

See [“Configuring the SIC Diameter Server \(SRC CLI\)”](#) on page 60.

2. Configure the Diameter application.

See [“Configuring the Diameter Application \(SRC CLI\)”](#) on page 137.

3. Configure the SRC Diameter server.

See [“Configuring Diameter Peers \(SRC CLI\)”](#) on page 143.

Configuring Management of RADIUS-Enabled Devices for the SIC (SRC CLI)

To configure management of RADIUS-enabled devices when using the SIC:

1. Configure the NAS group peers, device capabilities, and routes.

See [“Configuring the NAS Groups \(SRC CLI\)”](#) on page 145.

2. Configure the SAE to manage SAE devices.

See [“Configuring the SAE to Manage AAA Devices”](#) on page 150.

3. Configure the AAA policy rules.

See [“Configuring AAA Policies \(SRC CLI\)”](#) on page 152.

Configuring the Connection Between the SIC and the Juniper Networks Database (SRC CLI)

The configuration of the subscriber information collector (SIC) is stored in the Juniper Networks database.

Use the following statements to configure the connection between the SIC and the Juniper Networks database:

```
slot number sic initial directory-connection {  
    url url ;  
    port port ;  
    principal principal ;  
    credentials credentials ;  
    entry-dn entry-dn ;  
    filter filter ;  
}
```

To configure the directory connection properties for the SIC:

1. From configuration mode, access the statement that configures the directory configuration for the SIC in a slot.

```
user@host# edit slot number sic initial directory-connection
```

For example:

```
user@host# edit slot 0 sic initial directory-connection
```

2. Specify the password with which the SIC accesses the directory.

```
[edit slot 0 sic initial directory-connection]
user@host# set credentials credentials
```

3. (Optional) Specify the URL that identifies the location of the primary directory server.

```
[edit slot 0 sic initial directory-connection]
user@host# set url url
```

On a C Series Controller, this value is [ldap://127.0.0.1:389](#).

4. (Optional) Specify the port to use when connecting to the Juniper Networks database.

```
[edit slot 0 sic initial directory-connection]
user@host# set port port
```

5. (Optional) Specify the DN that contains the username that the directory server uses to authenticate the SIC.

```
[edit slot 0 sic initial directory-connection]
user@host# set principal principal
```

6. (Optional) Specify where the root of the SIC configuration is in the directory.

```
[edit slot 0 sic initial directory-connection]
user@host# set entry-dn entry-dn
```

7. (Optional) Specify any query filters you want to use to monitor changes in the Juniper Networks database.

```
[edit slot 0 sic initial directory-connection]
user@host# set filter filter
```

8. (Optional) Verify your configuration.

```
[edit slot 0 sic initial directory-connection]
user@host# show
url ldap://127.0.0.1:389/;
principal cn=conf,o=0perators,<base>;
credentials *****;
```

Related Documentation

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 155](#)
- [Configuring Initial Directory Eventing Properties for SRC Components](#)

- *Verifying the Local Configuration for a Component*

Creating an SIC Group and Server (SRC CLI)

The SIC group configuration controls the properties of accounting and authentication targets, dictionaries, editing rules, and RADIUS and Diameter options.

To create an SIC group and the associated server:

- From configuration mode, access the configuration statement that creates an SIC group.

```
[edit]
user@host# edit slot 0 server
```

For example, if you want to create an SIC group named `server-group1` that includes a server named `server-bldg5`, from configuration mode:

- Specify the *group-name* and *server-name*.

```
[edit]
user@host# edit slot 0 sic server
set name /server-group1/server-bldg5
```

The following rules depict how a new SIC group or server configuration is created on successfully committing the configuration:

- If the **group-name** does not exist in the Juniper Networks database, a new group and server instance as specified in this statement are created and populated with sample data.
- If the **group-name** already exists in the Juniper Networks database, a server instance as specified in this statement is created under the group and populated with sample data.

If you want to add another server to `server-group1` named `server-bldg5a`, execute:

```
[edit]
user@host# edit slot 0 sic server
set name /server-group1/server-bldg5a
```

Creating a server by using this statement populates it with sample data. You can also add a new server to an existing group by using the shared SIC group **shared sic group identifier server identifier** statement. However, this statement does not populate the server with sample data.

Related Documentation

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 155](#)
- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 12](#)
- [Creating an SIC Server Instance \(SRC CLI\) on page 51](#)

Creating an SIC Server Instance (SRC CLI)

Use either of the following statements to configure an SIC server instance:

```
slot number sic server {
    name /group-name/server-name;
}
shared sic group identifier server identifier
```

To create an instance of an SIC server:

- From configuration mode, access the statement that configures the SIC server.

```
[edit]
user@host# edit slot 0 sic server
set name /group-name/server-name
```

For example, if you want to create an SIC group named **server-group1** that includes a server named **server-bldg5**, from configuration mode:

- Specify the *group-name* and *server-name*.

```
[edit]
user@host# edit slot 0 sic server
set name /server-group1/server-bldg5
```

If you want to add another server to **server-group1** named **server-bldg5a**, execute:

```
[edit]
user@host# edit slot 0 sic server
set name /server-group1/server-bldg5a
```

Creating a server by using this statement populates it with sample data.

You can also add a new server to an existing group by using the **shared sic group *identifier* server *identifier*** statement. However, this statement does not populate the server with sample data.

Related Documentation

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 155](#)
- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 12](#)
- [Creating an SIC Group and Server \(SRC CLI\) on page 50](#)
- [Configuring Event Logging for an SIC Server \(SRC CLI\) on page 88](#)

Configuring Dictionaries for the SIC Group (SRC CLI)

You can add new attributes or modify the current attributes in an SIC dictionary. To add or modify an attribute in a dictionary, specify the unique name of the attribute and configure the RADIUS properties of the attribute.



NOTE: To create a new dictionary, we recommend that you work with Juniper Networks Technical Support.

Use the following statements to configure attributes in an SIC dictionary:

```
shared sic group identifier dictionary id
```

```
shared sic group identifier dictionary id attribute id
```

```
shared sic group identifier dictionary id attribute id radius {
    type type;
    format (one-byte-integer | integer | eight-byte-integer | string | ipv4-address |
        ipv6-address | time | octets);
    vendor-id vendor-id;
    encrypt;
    salt-encrypt;
    tagged;
    sensitive;
}
```

```
shared sic group identifier dictionary id attribute id radius constant constant-name {
    constant-value;
}
```

To add or modify attributes in an SIC dictionary:

1. From configuration mode, access the statement that specifies the unique name for the dictionary. This sample procedure uses `group1` as the group identifier and `dic1` as the dictionary identifier.

```
[edit]
user@host# edit shared sic group group1 dictionary dic1
```

2. Specify the unique name for the attribute you want to add or modify in the dictionary.

```
[edit shared sic group group1 dictionary dic1]
user@host# edit attribute id
```

3. Specify that the attribute is a RADIUS attribute.

```
[edit shared sic group group1 dictionary dic1 attribute id]
user@host# edit radius
```

4. Specify the attribute type.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set type type
```

5. Specify the format of the RADIUS attribute.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set format (one-byte-integer | integer | eight-byte-integer | string |
    ipv4-address | ipv6-address | time | octets)
```

where:

- one-byte-integer—Attribute value is an 8-bit unsigned integer.
- integer—Attribute value is a 32-bit unsigned integer.
- eight-byte-integer—Attribute value is a 64-bit unsigned integer.
- string—Attribute value is a string.
- ipv4-address—Attribute value is an IPv4 address.
- ipv6-address—Attribute value is an IPv6 address.
- time—Attribute value is a 32-bit unsigned value, with the most significant octet appearing first. The value is equal to the number of seconds since 00:00:00 UTC, January 1, 1970.
- octets—Attribute value consists of raw bytes.

6. (Optional) Specify the vendor ID for the attribute.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set vendor-id vendor-id
```

7. (Optional) Specify whether the attribute should be encrypted without the salt.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set encrypt
```

8. (Optional) Specify whether the attribute should be encrypted with the salt.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set salt-encrypt
```

9. (Optional) Specify whether the RADIUS attribute is tagged.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set tagged
```

10. (Optional) Specify whether the RADIUS attribute carries sensitive data, so its value will not be logged.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set sensitive
```

11. (Optional) Specify the name and value of the constant you want to associate with the data contained in the RADIUS attribute.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius constant]
user@host# set constant-name constant-name constant-value
```

12. (Optional) If you modify an existing dictionary, you need to restart the SIC.

```
user@host# restart component sic
```

- Related Documentation**
- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 155](#)
 - [Configuring the Device Models Supported by the SIC Group \(SRC CLI\) on page 54](#)
 - [SIC Dictionaries and Device Models Overview \(SRC CLI\) on page 28](#)

Configuring the Device Models Supported by the SIC Group (SRC CLI)

To configure the device models supported by the SIC group:

1. From configuration mode, access the statement that configures the device models supported by the SIC group. For example, to configure the device associated with the model name `dm1` for the group `group1`:

```
[edit]
user@host# edit shared sic group group1 model dm1
```

2. Specify the name of the dictionary used by the device model.

```
[edit shared sic group group1 model dm1]
user@host# set dictionary dictionary
```

- Related Documentation**
- [SIC Dictionaries and Device Models Overview \(SRC CLI\) on page 28](#)
 - [Configuring Dictionaries for the SIC Group \(SRC CLI\) on page 51](#)
 - [Configuring Upstream and Downstream RADIUS Network Elements \(SRC CLI\) on page 64](#)
 - [Example: Basic SIC Group Configuration \(SRC CLI\) on page 155](#)

Configuring the RADIUS Accounting Listener for the SIC Group (SRC CLI)

The accounting listener listens for RADIUS accounting events and filters undesired events based on attachment session attributes. Complete the following tasks to configure the accounting listener:

1. [Configuring the RADIUS Accounting Listener Queue Limits \(SRC CLI\) on page 54](#)
2. [Configuring the RADIUS Accounting Listener Transport \(SRC CLI\) on page 55](#)

Configuring the RADIUS Accounting Listener Queue Limits (SRC CLI)

Use the following statements to configure the accounting listener queue limits:

```
shared sic group identifier radius accounting-listener limit {
    incoming-queue incoming-queue;
    transaction-queue transaction-queue;
}
```

To configure the RADIUS accounting listener queue limits:

1. From configuration mode, access the statement that configures the RADIUS accounting listener queue limits. For example, to configure the limits for a group called `group1`:

```
[edit]
user@host# edit shared sic group group1 radius accounting-listener limit
```

2. (Optional) Specify the incoming queue limit for the RADIUS accounting listener.

```
[edit shared sic group group1 radius accounting-listener limit]
user@host# set incoming-queue incoming-queue
```

3. (Optional) Specify the transaction queue limit for the RADIUS accounting listener.

```
[edit shared sic group group1 radius accounting-listener limit]
user@host# set transaction-queue transaction-queue
```

Configuring the RADIUS Accounting Listener Transport (SRC CLI)

Use the following statements to configure the RADIUS accounting listener transport:

```
shared sic group identifier radius accounting-listener transport
shared sic group identifier radius accounting-listener transport id {
  port port;
  connections-per-thread connections-per-thread;
  connect-timeout connect-timeout;
  disconnect-timeout disconnect-timeout;
}
```

1. From configuration mode, access the statement that configures the RADIUS accounting listener transport and specify a name for the transport. Each RADIUS accounting transport must have a unique name. For example to configure a transport called `acct-tran1`:

```
[edit]
user@host# edit shared sic group group1 radius accounting-listener transport acct-tran1
```

2. Specify the UDP port number of the accounting listener from which the server listens for RADIUS packets.

```
[edit shared sic group group1 radius accounting-listener transport acct-tran1]
user@host# set port port
```

3. (Optional) Specify the number of UDP connections per thread.

```
[edit shared sic group group1 radius accounting-listener transport acct-tran1]
user@host# set connections-per-thread connections-per-thread
```

4. (Optional) Specify the UDP connection timeout in milliseconds.

```
[edit shared sic group group1 radius accounting-listener transport acct-tran1]
user@host# set connect-timeout connect-timeout
```

5. (Optional) Specify the UDP disconnection timeout in milliseconds.

```
[edit shared sic group group1 radius accounting-listener transport acct-tran1]
user@host# set disconnect-timeout disconnect-timeout
```

**Related
Documentation**

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 155](#)
- [Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\) on page 58](#)
- [Configuring the RADIUS Transport for an SIC Server \(SRC CLI\) on page 59](#)

Configuring the RADIUS Authentication Listener for the SIC Group (SRC CLI)

The authentication listener listens for RADIUS authentication messages and filters undesired events based on attachment session attributes. Complete the following tasks to configure the authentication listener:

1. [Configuring the RADIUS Authentication Listener Queue Limits \(SRC CLI\) on page 56](#)
2. [Configuring the RADIUS Authentication Listener Transport \(SRC CLI\) on page 57](#)

Configuring the RADIUS Authentication Listener Queue Limits (SRC CLI)

Use the following statements to configure the RADIUS authentication listener queue limit:

```
shared sic group identifier radius authentication-listener limit {
  incoming-queue incoming-queue;
  transaction-queue transaction-queue;
}
```

To configure the RADIUS authentication listener queue limits:

1. From configuration mode, access the statement that configures the RADIUS authentication listener queue limits. For example, to configure the limits for a group called group1:

```
[edit]
user@host# edit shared sic group group1 radius authentication-listener limit
```

2. (Optional) Specify the incoming queue limit for the RADIUS authentication listener.

```
[edit shared sic group group1 radius authentication-listener limit]
user@host# set incoming-queue incoming-queue
```

3. (Optional) Specify the transaction queue limit for the RADIUS authentication listener.

```
[edit shared sic group group1 radius authentication-listener limit]
user@host# set transaction-queue transaction-queue
```

Configuring the RADIUS Authentication Listener Transport (SRC CLI)

Use the following statements to configure the RADIUS authentication listener transport:

```
shared sic group identifier radius authentication-listener transport
shared sic group identifier radius authentication-listener transport id {
  port port;
  connections-per-thread connections-per-thread;
  connect-timeout connect-timeout;
  disconnect-timeout disconnect-timeout;
}
```

1. From configuration mode, access the statement that configures the RADIUS authentication listener transport and specify a name for the transport. Each RADIUS authentication transport must have a unique name. For example, to configure a transport called `auth-tran1`:

```
[edit]
user@host# edit shared sic group group1 radius authentication-listener transport
auth-tran1
```

2. Specify the UDP port number of the authentication listener from which the server listens for RADIUS packets.

```
[edit shared sic group group1 radius authentication-listener transport auth-tran1]
user@host# set port port
```

3. (Optional) Specify the number of UDP connections per thread.

```
[edit shared sic group group1 radius authentication-listener transport auth-tran1]
user@host# set connections-per-thread connections-per-thread
```

4. (Optional) Specify the UDP connection timeout in milliseconds.

```
[edit shared sic group group1 radius authentication-listener transport auth-tran1]
user@host# set connect-timeout connect-timeout
```

5. (Optional) Specify the UDP disconnection timeout in milliseconds.

```
[edit shared sic group group1 radius authentication-listener transport auth-tran1]
user@host# set disconnect-timeout disconnect-timeout
```

Related Documentation

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 155](#)
- [Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\) on page 58](#)
- [Configuring the RADIUS Transport for an SIC Server \(SRC CLI\) on page 59](#)

Configuring the Outbound RADIUS Transport of the SIC Group (SRC CLI)

You can use the RADIUS outbound transport to control communication to accounting targets that reside in a downstream network element. You need to specify the UDP port, as well as connect and disconnect related configuration options of the SIC group.

Use the following statements to configure the outbound RADIUS transport of the SIC group:

```
shared sic group identifier radius outbound-transport transport-name
```

```
shared sic group identifier radius outbound-transport transport-name {  
  connections-per-thread connections-per-thread;  
  connect-timeout connect-timeout;  
  disconnect-timeout disconnect-timeout;  
  port port;  
  port-range-size port-range-size;  
}
```

To configure the outbound RADIUS transport of the SIC group:

1. From configuration mode, access the statement that configures the name for the outbound RADIUS transport of the SIC group. For example, to configure the outbound RADIUS transport called outtrp1 for the SIC group group1:

```
[edit]  
user@host# edit shared sic group group1 radius outbound-transport outtrp1
```

2. (Optional) Specify the number of UDP connections per thread.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set connections-per-thread connections-per-thread
```

3. (Optional) Specify the UDP connection timeout in milliseconds.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set connect-timeout connect-timeout
```

4. (Optional) Specify the UDP disconnection timeout in milliseconds.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set disconnect-timeout disconnect-timeout
```

5. Specify the UDP port number starting from which the server sends the RADIUS packets.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set port port
```

6. (Optional) Specify the range of UDP ports that are used to send the RADIUS packets.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set port-range-size port-range-size
```

- Related Documentation**
- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 155](#)
 - [RADIUS and Diameter Transports on page 27](#)
 - [Configuring the RADIUS Transport for an SIC Server \(SRC CLI\) on page 59](#)
 - [Configuring Downstream Network Elements and Accounting and Authentication Targets \(SRC CLI\) on page 70](#)

Configuring the RADIUS Transport for an SIC Server (SRC CLI)

You need to configure both the inbound and outbound RADIUS transport for each server in the SIC group. Servers use the same inbound and outbound transport names that are configured for the SIC group.

Use the following statements to configure the RADIUS transport options for the SIC server:

```
shared sic group identifier server identifier transport transport-name
```

```
shared sic group identifier server identifier transport transport-name {
  address address;
}
```

```
shared sic group identifier server identifier outbound-transport transport-name
```

```
shared sic group identifier server identifier outbound-transport transport-name {
  address address;
}
```

To configure the RADIUS transport options for the SIC server:

1. From configuration mode, access the statement that configures the RADIUS inbound transport options for the server. For example, if the accounting listener transport for the group is configured as `trpin1`, specify the server inbound transport as `trpin1`.

```
[edit]
user@host# edit shared sic group group1 server server1 transport trpin1
```

2. (Optional) Specify the IP address used by the server for receiving UDP packets.

```
[edit shared sic group group1 server server1 transport trpin1]
user@host# set address address
```

3. Specify the RADIUS outbound transport options for the SIC server. For example, if the outbound transport for the SIC group is set to `trpout1`, set the server outbound transport to `trpout1`.

```
[edit]
user@host# edit shared sic group group1 server server1 outbound-transport trpout1
```

4. (Optional) Specify the IP address used by the server when sending outbound requests.

```
[edit shared sic group group1 server server1 outbound-transport trpout1]
user@host# set address address
```

**Related
Documentation**

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 155](#)
- [RADIUS and Diameter Transports on page 27](#)
- [Managing Dynamic Services on RADIUS-Enabled Devices on page 35](#)
- [Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\) on page 58](#)
- [Creating an SIC Server Instance \(SRC CLI\) on page 51](#)

Configuring the SIC Diameter Server (SRC CLI)

- [Configuration Statements for the SIC Diameter Server \(SRC CLI\) on page 60](#)
- [Configuring the SIC Diameter Server Identity \(SRC CLI\) on page 61](#)
- [Configuring the SIC Diameter Server Peer \(SRC CLI\) on page 62](#)

Configuration Statements for the SIC Diameter Server (SRC CLI)

Use the following statements to configure the SIC Diameter server:

```
shared sic group identifier server identifier diameter identity {
    origin-host origin-host;
    origin-realm origin-realm;
}
shared sic group identifier server identifier diameter transport id {
    protocol (tcp | sctp);
    port port;
}
shared sic group identifier diameter network-element id {
    description description;
    failover-policy (round-robin | primary-backup);
}
shared sic group identifier diameter network-element id peer name {
    description description;
    address address;
    protocol (tcp | sctp);
    port port;
    active-peer;
    priority priority;
}
shared sic group identifier diameter network-element id peer name {
    enforce-source-address;
}
shared sic group identifier diameter network-element id peer name {
    origin-host origin-host;
}
shared sic group identifier diameter network-element id peer name addresses address
address
```

Configuring the SIC Diameter Server Identity (SRC CLI)

Configuring the SIC Diameter server identity includes specifying the origin-host, origin-realm, the port the server receives Diameter messages on, and protocol. The SIC Diameter server communicates with the SRC Diameter server. The origin-host and origin-realm identify the SIC Diameter server. This identity is sent in all Diameter requests originating on this server.

The default identity of the SIC Diameter server is set to origin-host="your-host" and the origin-realm="your-realm.net." You must reconfigure these settings for your network environment.

To configure the SRC Diameter server and the Diameter application, see ["Configuring the Diameter Application \(SRC CLI\)" on page 137](#) and ["Configuring Diameter Peers \(SRC CLI\)" on page 143](#).

Use the following statements to configure the SIC Diameter server identity:

```
shared sic group identifier server identifier diameter identity {
  origin-host origin-host;
  origin-realm origin-realm;
}
shared sic group identifier server identifier diameter transport id {
  protocol (tcp | sctp);
  port port;
}
```

To configure the SIC Diameter server identity:

1. From configuration mode, access the statement that configures the SIC Diameter server. For example, to configure the SIC Diameter server in an SIC group called g1 that includes an SIC server called svr1:

```
[edit]
user@host# shared sic group g1 server svr1 diameter identity
```

2. Specify the origin-host name of the SIC Diameter server. For example, to specify the origin-host as sic-diam-svr1:

```
[edit shared sic group g1 server svr1 diameter identity]
user@host# set origin-host sic-diam-svr1
```

3. Specify the origin-realm name of the SIC Diameter server. For example, to specify the origin-realm as abc.com:

```
[edit shared sic group g1 server svr1 diameter identity]
user@host# set origin-realm abc.com
```

4. Verify your configuration.

```
[edit shared sic group g1 server svr1 diameter identity]
user@host# show
```

```
user@host# show
origin-host diam-svr1;
origin-realm abc.com;
```

Configuring the SIC Diameter Server Peer (SRC CLI)

The SIC Diameter server handles all communication between the SIC and the SRC Diameter server. This procedure describes how to configure the network element in which the SRC Diameter server logically resides, the failover policy, and the Diameter connection between the SIC Diameter server and the SRC Diameter server.

Use the following statements to configure the SIC Diameter peer:

```
shared sic group identifier diameter network-element id {
  description description;
  failover-policy (round-robin | primary-backup);
}
shared sic group identifier diameter network-element id peer name {
  description description;
  address address;
  protocol (tcp | sctp);
  port port;
  active-peer;
  priority priority;
}
shared sic group identifier diameter network-element id peer name {
  enforce-source-address;
}
shared sic group identifier diameter network-element id peer name {
  origin-host origin-host;
}
shared sic group identifier diameter network-element id peer name addresses address
address
```

To configure the SIC Diameter server peer:

1. From configuration mode, access the statement that configures the SIC Diameter server peer and configure the network element where the SRC Diameter server resides. For example, to configure a Diameter network element called `diam-ne1` for an SIC group called `g1`:

```
[edit]
user@host# shared sic group g1 diameter network-element diam-ne1
```

2. (Optional) Specify a description for the network element.

```
[shared sic group g1 diameter network-element diam-ne1]
user@host# set description description
```

3. (Optional) Configure the failover policy for the network element. For example, to configure the primary or backup failover policy:

```
[shared sic group g1 diameter network-element diam-ne1]
user@host# set primary-backup
```


4. Configure the name of the Diameter peer (SRC Diameter server). For example, to call the peer `src-diam-svr1`:

```
[shared sic group g1 diameter network-element diam-ne1]
user@host# edit peer src-diam-svr1
```

5. (Optional) Specify a description for the Diameter peer.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set description description
```

6. Specify the IP address of the remote Diameter peer (SRC Diameter server). For example, `10.1.2.3`.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set address 10.1.2.3
```

7. Specify the protocol the Diameter peer (SRC Diameter server) uses for Diameter messages (TCP or SCTP).

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set protocol sctp
```

8. Specify which port the Diameter peer (SRC Diameter server) receives messages on. For example, port `2222`.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set port 2222
```

9. (Optional) Specify whether the peer is active or not. If the peer is configured to connect actively, the server periodically attempts to connect (or reconnect after a connection has failed) to the remote peer. If this option is not set, a connection is established only after the remote peer attempts to connect to this server.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set active-peer
```

10. (Optional) Specify the priority of the peer for the failover policy. Peers with lower priority values are the preferred routing targets for Diameter requests. Requests are split equally among peers with the same priority level.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set priority 1
```

11. (Optional) Specify whether a source IP match is required for the connection. This option determines whether the source IP address of a connection attempt must match one of the configured IP addresses used to connect to this peer. If this option is not set, requests are accepted from any IP address as long as the client presents the correct host name during the capabilities exchange. This functionality allows other peers to exist behind NAS devices.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set enforce-source-address
```

12. Specify the origin-host name of the Diameter peer (SRC Diameter server). For example, if the origin-host name of the SRC Diameter server is `diam-host1`:

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set origin-host diam-host1
```

13. (Optional) Specify an ordered set of IP addresses to use for a multilink connection. An IP address of the remote peer is necessary to establish a Diameter connection with the remote peer (SRC Diameter server). For a Diameter connection over TCP, only one configured address is used. Over SCTP, the connection may be established over multiple addresses.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set addresses address 10.1.2.4
user@host# set addresses address 10.1.2.5
user@host# set addresses address 10.1.2.6
```

14. Verify your configuration.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# show
```

```
active-peer;
address 10.1.2.3;
addresses {
  address 10.1.2.4;
  address 10.1.2.5;
  address 10.1.2.6;
}
port 3868;
priority 1;
protocol sctp;
enforce-source-address;
origin-host diam-host1;
```

```
[edit shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host#
```

Configuring Upstream and Downstream RADIUS Network Elements (SRC CLI)

- [Configuration Statements for Downstream Network Elements and Accounting and Authentication Targets \(SRC CLI\) on page 65](#)
- [Configuration Statements for Upstream Network Elements, Accounting and Authentication Clients, and Dynamic Authorization Targets \(SRC CLI\) on page 66](#)
- [Creating a Network Element \(SRC CLI\) on page 66](#)
- [Configuring the Device Models Supported in the Network Element \(SRC CLI\) on page 67](#)
- [Configuring Upstream Network Elements and Accounting and Authentication Clients \(SRC CLI\) on page 68](#)
- [Configuring Upstream Network Elements and Dynamic Authorization Targets \(SRC CLI\) on page 69](#)

- [Configuring Downstream Network Elements and Accounting and Authentication Targets \(SRC CLI\) on page 70](#)
- [Configuration Statements for SIC Group Failover Mode and Policy \(SRC CLI\) on page 72](#)
- [Configuring Failover Mode and Policy \(SRC CLI\) on page 73](#)

Configuration Statements for Downstream Network Elements and Accounting and Authentication Targets (SRC CLI)

Use the following statements to configure downstream RADIUS network elements and accounting and authentication targets for the SIC group:

```
shared sic group identifier radius network-element id
```

```

shared sic group identifier radius network-element id downstream {
  model model;
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) {
  failover-mode (round-robin | primary-backup);
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) failover-policy
shared sic group identifier radius network-element id downstream (authentication |
  accounting) failover-policy fast-fail {
  minimum-number minimum-number;
  timeout timeout;
  reset-delay reset-delay;
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) failover-policy retry {
  number number;
  timeout timeout;
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) accounting-target name {
  address address;
  priority priority;
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) accounting-target name {
  secret secret;
  outbound-transport outbound-transport;
  port port;
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) authentication-target name {
  address address;
  priority priority;
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) authentication-target name {
  secret secret;
  outbound-transport outbound-transport;
  port port;

```

```
}
```

Configuration Statements for Upstream Network Elements, Accounting and Authentication Clients, and Dynamic Authorization Targets (SRC CLI)

Use the following statements to configure upstream RADIUS network elements, accounting and authentication clients, and dynamic authorization targets for the SIC group:

```
shared sic group identifier radius network-element id
shared sic group identifier radius network-element id upstream {
    model model;
}
shared sic group identifier radius network-element id upstream radius-client id {
    address address;
    accounting-secret accounting-secret;
    authentication-secret authentication-secret;
}
shared sic group identifier radius network-element id upstream dynamic-authorization-target
{
    failover-mode (round-robin | primary-backup);
}
shared sic group identifier radius network-element id upstream dynamic-authorization-target
    failover-policy
shared sic group identifier radius network-element id upstream dynamic-authorization-target
    failover-policy retry {
        number number;
        timeout timeout;
    }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
    failover-policy fast-fail {
        minimum-number minimum-number;
        timeout timeout;
        reset-delay reset-delay;
    }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
    target name {
        address address;
        priority priority;
    }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
    target name {
        secret secret;
        port port;
    }
}
```

Creating a Network Element (SRC CLI)

Network elements are logical entities that are considered either upstream or downstream from the SIC. Upstream network elements contain logical clients and targets for NAS devices. Downstream network elements contain logical targets for the downstream AAA server responsible for accounting and authentication.

Use the following statement to create a network element:

shared sic group *identifier* radius network-element *id*

To create a network element:

- From configuration mode, access the statement that creates a RADIUS network element. For example, to create a network element called ne1 for the SIC group group1:

```
[edit]
user@host# edit shared sic group group1 radius network-element ne1
```

Configuring the Device Models Supported in the Network Element (SRC CLI)

You must configure which device models are supported by the upstream and downstream network elements.



NOTE: To assign a device model to a network element, you must first configure the device models and the associated dictionaries supported by the SIC group using the shared sic group *identifier* model *id* statement. See “Configuring the Device Models Supported by the SIC Group (SRC CLI)” on page 54.

Use the following statements to configure the device model:

```
shared sic group identifier radius network-element id downstream {
    model model;
}
shared sic group identifier radius network-element id upstream {
    model model;
}
```

To configure the device models supported in the network element:

- From configuration mode, access the statement that configures the RADIUS network element and specify a name for the network element. This sample procedure uses group1 for the SIC group and ne1 for the downstream network element identifier.

```
[edit]
user@host# edit shared sic group group1 radius network-element ne1 downstream
```

- Specify a device model. The device model must have previously been configured for the SIC group.

```
[edit shared sic group group1 radius network-element ne1 downstream]
user@host# set model model
```

Configuring Upstream Network Elements and Accounting and Authentication Clients (SRC CLI)

Accounting and authentication clients are NAS devices that logically reside in upstream network elements. Accounting clients send RADIUS accounting requests to the SIC accounting listener. Authentication clients send RADIUS authentication requests to the SIC authentication listener. You must configure at least one accounting client and one authentication client. Each client must have a unique name and address.

Use the following statements to configure accounting clients:

```
shared sic group identifier radius network-element id upstream radius-client id {  
    address address;  
    accounting-secret accounting-secret;  
    authentication-secret authentication-secret;  
}
```

To configure RADIUS accounting and authentication clients:

1. From configuration mode, access the statement that configures an upstream network element and RADIUS client. For example, to configure an upstream RADIUS network element called `ne1` and RADIUS client called `rc1` for the SIC group `group1`:

```
[edit]  
user@host# edit shared sic group group1 radius network-element ne1 upstream  
radius-client rc1
```

2. (Optional) Specify the IP address of the RADIUS client.

```
[edit shared sic group group1 radius network-element ne1 upstream radius-client rc1]  
user@host# set address address
```

3. (Optional) Specify the shared secret used by the accounting client.

```
[edit shared sic group group1 radius network-element ne1 upstream radius-client rc1]  
user@host# set accounting-secret authentication-secret
```

4. Specify the shared secret used by the authentication client.

```
[edit shared sic group group1 radius network-element ne1 upstream accounting-client]  
user@host# set accounting-secret accounting-secret
```

Configuring Upstream Network Elements and Dynamic Authorization Targets (SRC CLI)

Dynamic authorization targets are logical entities that represent the NAS device in upstream network elements. The SIC forwards COA/DM requests to dynamic authorization targets.

Use the following statements to configure dynamic authorization targets:

```
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  target name {
    address address;
    priority priority;
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  target name {
    secret secret;
    port port;
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  {
    failover-mode (round-robin | primary-backup);
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  failover-policy {
    priority priority;
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  failover-policy retry {
    number number;
    timeout timeout;
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  failover-policy fast-fail {
    minimum-number minimum-number;
    timeout timeout;
    reset-delay reset-delay;
  }
```

To configure a dynamic authorization target:

1. From configuration mode, access the statement that configures an upstream network element and dynamic authorization target. For example, to configure an upstream RADIUS network element called `ne1` and dynamic authorization target called `dat1` for the SIC group `group1`:

```
[edit]
user@host# edit shared sic group group1 radius network-element ne1 upstream
dynamic-authorization-target target dat1
```

2. Specify the IP address of the target.

```
[edit shared sic group group1 radius network-element ne1 upstream
dynamic-authorization-target target dat1]
user@host# set address address
```

3. Specify the priority of the target. Targets with lower priority values are selected before other targets in a failover policy.

```
[edit shared sic group group1 radius network-element ne1 upstream
dynamic-authorization-target target dat1]
user@host# set priority priority
```

4. Specify the shared secret used by the target.

```
[edit shared sic group group1 radius network-element ne1 upstream
dynamic-authorization-target target dat1]
user@host# set secret secret
```

5. (Optional) Specify the port used by the target to receive dynamic authorization messages.

```
[edit shared sic group group1 radius network-element ne1 upstream
dynamic-authorization-target target dat1]]
user@host# set port port
```

Configuring Downstream Network Elements and Accounting and Authentication Targets (SRC CLI)

Accounting and authentication targets (RADIUS AAA server) receive requests forwarded by the SIC. These targets reside in downstream network elements. You must configure at least one accounting target and one authentication target. Each target must have a unique name and address.

1. [Configuring SIC Accounting Targets \(SRC CLI\) on page 70](#)
2. [Configuring SIC Authentication Targets \(SRC CLI\) on page 71](#)

Configuring SIC Accounting Targets (SRC CLI)

Use the following statements to configure accounting targets:

```
shared sic group identifier radius network-element id downstream (authentication |
accounting) accounting-target name {
  address address;
  priority priority;
}
shared sic group identifier radius network-element id downstream (authentication |
accounting) accounting-target name {
  secret secret;
  outbound-transport outbound-transport;
  port port;
}
```

To configure an accounting target:

1. From configuration mode, access the statement that configures the accounting target. This sample procedure uses group1 for the group identifier, ne1 for the network element identifier, and target1 as the accounting target name.


```
edit shared sic group group1 radius network-element ne1 downstream accounting
accounting-target target1
```

- Specify the IP address of the RADIUS accounting target contained in the network element.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
accounting-target target1]
user@host# set address address
```

- Specify the priority of the target. Targets with lower priority values are selected before other targets in a failover policy.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
accounting-target target1]
user@host# set priority priority
```

- Specify the shared secret used by the RADIUS accounting target.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
accounting-target target1]
user@host# set secret secret
```

- (Optional) Specify the name of the local transport used to send requests to the accounting target.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
accounting-target target1]
user@host# set outbound-transport outbound-transport
```

- (Optional) Specify the UDP port number on which the RADIUS accounting target listens for requests.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
accounting-target target1]
user@host# set port port
```

Configuring SIC Authentication Targets (SRC CLI)

Use the following statements to configure authentication targets:

```
shared sic group identifier radius network-element id downstream (authentication |
accounting) authentication-target name {
  address address;
  priority priority;
}
shared sic group identifier radius network-element id downstream (authentication |
accounting) authentication-target name {
  secret secret;
  outbound-transport outbound-transport;
  port port;
}
```

To configure an authentication target:

1. From configuration mode, access the statement that configures the authentication target. This sample procedure uses `group1` for the group identifier, `ne1` for the network element identifier, and `target1` as the authentication target name.

```
edit shared sic group group1 radius network-element ne1 downstream authentication  
authentication-target target1
```

2. Specify the IP address of the RADIUS authentication target contained in the network element.

```
[edit shared sic group group1 radius network-element ne1 downstream authentication  
authentication-target target1]  
user@host# set address address
```

3. Specify the priority of the target. Targets with lower priority values are selected before other targets in a failover policy.

```
[edit shared sic group group1 radius network-element ne1 downstream authentication  
authentication-target target1]  
user@host# set priority priority
```

4. Specify the shared secret used by the RADIUS authentication target.

```
[edit shared sic group group1 radius network-element ne1 downstream authentication  
authentication-target target1]  
user@host# set secret secret
```

5. (Optional) Specify the name of the local transport used to send outbound requests to the authentication target.

```
[edit shared sic group group1 radius network-element ne1 downstream authentication  
authentication-target target1]  
user@host# set outbound-transport outbound-transport
```

6. (Optional) Specify the UDP port number on which the RADIUS authentication target listens for requests.

```
[edit shared sic group group1 radius network-element ne1 downstream authentication  
authentication-target target1]  
user@host# set port port
```

Configuration Statements for SIC Group Failover Mode and Policy (SRC CLI)

Use the following statements to configure failover mode and policy:

```
shared sic group identifier radius network-element id downstream (authentication |  
accounting) {  
    failover-mode (round-robin | primary-backup);  
}  
shared sic group identifier radius network-element id downstream (authentication |  
accounting) failover-policy  
shared sic group identifier radius network-element id downstream (authentication |  
accounting) failover-policy fast-fail {  
    minimum-number minimum-number;
```

```

        timeout timeout;
        reset-delay reset-delay;
    }
    shared sic group identifier radius network-element id downstream (authentication |
        accounting) failover-policy retry {
        number number;
        timeout timeout;
    }
    shared sic group identifier radius network-element id upstream dynamic-authorization-target
    {
        failover-mode (round-robin | primary-backup);
    }
    shared sic group identifier radius network-element id upstream dynamic-authorization-target
        failover-policy
    shared sic group identifier radius network-element id upstream dynamic-authorization-target
        failover-policy retry {
        number number;
        timeout timeout;
    }
    shared sic group identifier radius network-element id upstream dynamic-authorization-target
        failover-policy fast-fail {
        minimum-number minimum-number;
        timeout timeout;
        reset-delay reset-delay;
    }
}

```

Configuring Failover Mode and Policy (SRC CLI)

You must configure failover mode and policy for accounting and authentication targets upstream by completing the following tasks:

1. [Configuring Failover Mode \(SRC CLI\) on page 73](#)
2. [Configuring Fast Fail Options for the Failover Policy on page 74](#)
3. [Configuring Retry Options for the Failover Policy on page 75](#)

Configuring Failover Mode (SRC CLI)

You must configure failover mode for both accounting and authentication messages. Use the following statement to configure failover mode:

```

shared sic group identifier radius network-element id downstream (authentication |
    accounting) {
    failover-mode (round-robin | primary-backup);
}

```

To configure failover mode:

1. From configuration mode, access the statement that configures the network element failover mode and specify whether the connection is for authentication or accounting messages.

For example, this sample procedure uses `group1` for the group identifier, `ne1` for the network element identifier, and `accounting` as the connection.

[edit]

```
user@host# edit shared sic group group1 radius network-element ne1 downstream  
accounting
```

2. Specify failover mode used by the network element.

```
[edit shared sic group group1 radius network-element ne1 downstream]  
user@host# set failover-mode (round-robin | primary-backup)
```

Where:

- **round-robin**—When this failover mode is used, messages are sent to the network element over alternating paths.
- **primary-backup**—When this failover mode is used, messages are sent over the primary path unless it is unavailable, in which case messages are sent over the backup path.

Configuring Fast Fail Options for the Failover Policy

You must configure fast fail options for the failover policy for both accounting and authentication messages. Use the following statement to configure fast fail options:

```
shared sic group identifier radius network-element id downstream (authentication |  
accounting) failover-policy  
shared sic group identifier radius network-element id downstream (authentication |  
accounting) failover-policy fast-fail {  
  minimum-number minimum-number;  
  timeout timeout;  
  reset-delay reset-delay;  
}
```

To configure fast fail options for the failover policy:

1. From configuration mode, access the statement that configures fast fail options for the failover policy. For example, this sample procedure uses `group1` for the group identifier, `ne1` for the network element identifier, and `accounting` as the connection type.

```
edit shared sic group group1 radius network-element ne1 downstream accounting  
failover-policy fast-fail
```

2. Specify the minimum number of times the message is retransmitted if an acknowledgment from the target is not received.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting  
failover-policy fast-fail]  
user@host# set minimum-number minimum-number
```

3. Specify the time in seconds before the target is placed into fast fail mode.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting  
failover-policy fast-fail]  
user@host# set timeout timeout
```

- Specify the time in seconds after which the target is taken out of fast fail mode.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
 failover-policy fast-fail]
user@host# set reset-delay reset-delay
```

Configuring Retry Options for the Failover Policy

You must configure retry options for the failover policy for both accounting and authentication messages. Use the following statement to configure retry options:

```
shared sic group identifier radius network-element id downstream (authentication |
 accounting) failover-policy retry {
  number number;
  timeout timeout;
}
```

To configure retry options for the failover policy:

- From configuration mode, access the statement that configures retry options for the failover policy. For example, this sample procedure uses `group1` for the group identifier, `ne1` for the network element identifier, and `accounting` as the connection type.

```
edit shared sic group group1 radius network-element ne1 downstream accounting
 failover-policy retry
```

- Specify the maximum number of times a message is retransmitted if an acknowledgment from the target is not received.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
 failover-policy retry]
user@host# set number number
```

- Specify the number of seconds between retry attempts.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
 failover-policy retry]
user@host# set timeout timeout
```

Configuring What Realms Are Local to the SIC Group (SRC CLI)

To configure what realms are local to the SIC group:

- From configuration mode, access the statement that configures local realms. For example, to configure the local realm called `realm1` for the group `group1`:

```
[edit]
user@host# edit shared sic group group1 local-realm realm1
```

Related Documentation

- [SIC Local Realms Overview on page 29](#)
- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 12](#)

- [Creating an SIC Group and Server \(SRC CLI\) on page 50](#)
- [Creating an SIC Server Instance \(SRC CLI\) on page 51](#)

Configuration Statements for SIC Editing Rules (SRC CLI)

Use the following statements to configure the optional SIC editing rules at the **[edit]** hierarchy level.

Use the following statements to create the editing rule and specify the type of source used in the editing rule:

```
shared sic group identifier editing editing-rule {  
    mode (replace | append);  
}  
shared sic group identifier editing editing-rule default {  
    literal literal;  
    request-attribute request-attribute;  
    variable variable;  
}
```

Use the following statements to configure the editing rule when you specify a literal as the source of the editing rule:

```
shared sic group identifier editing editing-rule source literal  
shared sic group identifier editing editing-rule source literal identifier condition realm {  
    (present | not-present);  
}  
shared sic group identifier editing editing-rule source literal identifier condition realm  
    does-not-equal value  
shared sic group identifier editing editing-rule source literal identifier condition realm equals  
    value  
shared sic group identifier editing editing-rule source literal identifier condition realm  
    has-prefix value  
shared sic group identifier editing editing-rule source literal identifier condition realm  
    has-suffix value  
shared sic group identifier editing editing-rule source literal identifier condition realm range  
    {  
        low low;  
        high high;  
    }  
shared sic group identifier editing editing-rule source literal identifier condition request {  
}  
shared sic group identifier editing editing-rule source literal identifier condition request  
    attribute attribute-name {  
        (present | not-present);  
    }  
shared sic group identifier editing editing-rule source literal identifier condition request  
    attribute attribute-name does-not-equal value  
shared sic group identifier editing editing-rule source literal identifier condition request  
    attribute attribute-name equals value  
shared sic group identifier editing editing-rule source literal identifier condition request  
    attribute attribute-name has-prefix value
```

```

shared sic group identifier editing editing-rule source literal identifier condition request
  attribute attribute-name has-suffix value
shared sic group identifier editing editing-rule source literal identifier condition request
  attribute attribute-name range {
    low low;
    high high;
  }
shared sic group identifier editing editing-rule source literal identifier condition user-identity
  {
    (present | not-present);
  }
shared sic group identifier editing editing-rule source literal identifier condition user-identity
  does-not-equal value
shared sic group identifier editing editing-rule source literal identifier condition user-identity
  equals value
shared sic group identifier editing editing-rule source literal identifier condition user-identity
  has-prefix value
shared sic group identifier editing editing-rule source literal identifier condition user-identity
  has-suffix value
shared sic group identifier editing editing-rule source literal identifier condition user-identity
  range {
    low low;
    high high;
  }

```

Use the following statements to configure the editing rule when you specify a request attribute as the source of the editing rule:

```

shared sic group identifier editing editing-rule source request-attribute identifier {
  remove-prefix remove-prefix;
  remove-suffix remove-suffix;
  remove-before remove-before;
  remove-after remove-after;
}
shared sic group identifier editing editing-rule source request-attribute identifier condition
  realm {
    (present | not-present);
  }
shared sic group identifier editing editing-rule source request-attribute identifier condition
  realm does-not-equal value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  realm equals value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  realm has-prefix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  realm has-suffix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  realm range {
    low low;
    high high;
  }
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request {
  }
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request attribute attribute-name {
  }

```

```

    (present | not-present);
}
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request attribute attribute-name does-not-equal value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request attribute attribute-name equals value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request attribute attribute-name has-prefix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request attribute attribute-name has-suffix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request attribute attribute-name range {
    low low;
    high high;
  }
shared sic group identifier editing editing-rule source request-attribute identifier condition
  user-identity {
    (present | not-present);
  }
shared sic group identifier editing editing-rule source request-attribute identifier condition
  user-identity does-not-equal value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  user-identity equals value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  user-identity has-prefix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  user-identity has-suffix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  user-identity range {
    low low;
    high high;
  }
}

```

Use the following statements to configure the editing rule when you specify an SIC variable as the source of the editing rule:

```

shared sic group identifier editing editing-rule source variable identifier
shared sic group identifier editing editing-rule source variable identifier condition realm {
  (present | not-present);
}
shared sic group identifier editing editing-rule source variable identifier condition realm
  does-not-equal value
shared sic group identifier editing editing-rule source variable identifier condition realm
  equals value
shared sic group identifier editing editing-rule source variable identifier condition realm
  has-prefix value
shared sic group identifier editing editing-rule source variable identifier condition realm
  has-suffix value
shared sic group identifier editing editing-rule source variable identifier condition realm
  range {
    low low;
    high high;
  }
}
shared sic group identifier editing editing-rule source variable identifier condition request {
}
}

```



```

shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name {
    (present | not-present);
  }
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name does-not-equal value
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name equals value
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name has-prefix value
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name has-suffix value
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name range {
    low low;
    high high;
  }
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  {
    (present | not-present);
  }
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  does-not-equal value
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  equals value
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  has-prefix value
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  has-suffix value
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  range {
    low low;
    high high;
  }

```

Use the following statements to configure the target of the editing rule:

```

shared sic group identifier editing editing-rule target {
  request-attribute request-attribute;
  response-attribute response-attribute;
  variable variable;
}

```

Related Documentation

- [SIC Editing Rules \(SRC CLI\) on page 17](#)
- [Configuring the Optional Editing Rules Used by the SIC Group \(SRC CLI\) on page 80](#)
- [Configuring Explicit Routing \(SRC CLI\) on page 85](#)
- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 155](#)

Configuring the Optional Editing Rules Used by the SIC Group (SRC CLI)

When you use explicit routing for the SIC, you can optionally specify an editing rule you want applied to the accounting or authentication request before SIC sends the request to the target. To configure editing rules, you define a source, conditions, and a target.

Table 10 on page 80 lists the available sources, conditions, and targets you can define in editing rules, and “Configuration Statements for SIC Editing Rules (SRC CLI)” on page 76 provides a complete list of configuration statements used to define editing rules.

Table 10: SIC Editing Rule Options

Source	Conditions	Target
SIC literal	Match conditions:	Transactional variable
Transactional variable	<ul style="list-style-type: none"> • Realm • User identity 	RADIUS attribute in the request
RADIUS attribute in the request	<ul style="list-style-type: none"> • Request attribute 	RADIUS attribute in the response
	Condition tests:	
	<ul style="list-style-type: none"> • Present • Not present • Equals • Does not equal • Has suffix • Has prefix • Within range 	

To configure an editing rule:

1. From configuration mode, access the statement that configures the editing rule, and specify a name for the editing rule. For example, to create an editing rule called `er1`:

```
[edit]
user@host# edit shared sic group identifier editing er1
```

2. Specify the editing mode.

```
[edit shared sic group identifier editing er1]
user@host# set mode (replace | append)
```

Where:

- **replace**—Current target (LValue) is replaced with the new value from the editing process
- **append**—Current target (LValue) value is concatenated with the new target value from the editing process

3. Define the source of the editing rule. The source can be a literal, transactional variable, or an attribute in the request. For example, to define a literal called `literal1` as the source:

```
[edit]
edit shared sic group identifier editing er1 source literal
user@host# set literal1
```

4. (Optional) If the source is a request attribute, you can also specify whether to remove the prefix, remove the suffix, remove before @, or remove after @.

```
[edit ]
user@host# edit shared sic group identifier editing er1 source request-attribute identifier
user@host# set remove-prefix remove-prefix
```

5. Define the editing rule conditions, which include specifying the match conditions and the condition tests. See [Table 10 on page 80](#) and “[Configuration Statements for SIC Editing Rules \(SRC CLI\)](#)” on page 76 for a complete list of configuration statements used to specify SIC editing rules. For example, to specify a condition that examines literals in accounting requests for the realm=`abc.com`:

```
[edit]
edit shared sic group identifier editing er1 literal literal1 condition realm equals
user@host# set abc.com
```

6. Define the target (where you want the result of the editing process to be placed) of the editing rule. The target can be a transactional variable, a RADIUS attribute in the request, or a RADIUS attribute in the response. For example, to place the results of the editing process in a variable called `sic-variable1`:

```
[edit]
user@host# edit shared sic group identifier editing er1 target
user@host# set variable sic-variable1
```

7. (Optional) Specify a default editing rule. You can set default editing rules for all three source types (literal, variable, and request attribute).

```
[edit]
user@host# edit shared sic group identifier editing editing-rule default
user@host# set literal literal
```

Related Documentation

- [SIC Editing Rules \(SRC CLI\) on page 17](#)
- [Configuring Explicit Routing \(SRC CLI\) on page 85](#)
- [Accounting Method and Target \(SRC CLI\) on page 13](#)
- [Configuration Statements for SIC Editing Rules \(SRC CLI\) on page 76](#)
- [Configuration Statements for SIC Explicit Accounting Routing Rules on page 83](#)

- [Request Routing \(SRC CLI\) on page 14](#)
- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 155](#)

Configuring the Accounting Method Used by the SIC Group (SRC CLI)

- [Configuring Proxy RADIUS as the Accounting Method \(SRC CLI\) on page 82](#)

Configuring Proxy RADIUS as the Accounting Method (SRC CLI)

When you use the proxy RADIUS accounting method, the SIC forwards accounting messages to a remote AAA server (accounting target) located in a downstream network element for processing.

To use the proxy RADIUS accounting method, you need to have previously configured the downstream network element, associated the accounting target, and committed the configuration. For details about configuring downstream network elements and accounting targets, see [“Configuring Downstream Network Elements and Accounting and Authentication Targets \(SRC CLI\)” on page 70](#).

Use the following statements to configure the proxy accounting method:

```
shared sic group identifier accounting-method accounting-method-name
```

```
shared sic group identifier accounting-method accounting-method-name proxy radius {  
    network-element network-element;  
}
```

To configure proxy RADIUS as the accounting method for the SIC group:

1. From configuration mode, access the statement that configures the accounting method. For example, to configure an accounting method called acm2 for the SIC group group2 and specify proxy RADIUS as the accounting method:

```
[edit]  
user@host# edit shared sic group group2 accounting-method acm2 proxy radius
```

2. Specify the name of the previously configured downstream network element that contains the AAA server (accounting target) you want the SIC to forward accounting events to. For example, to forward accounting events to the downstream network element ne2:

```
[edit shared sic group group2 accounting-method acm2 proxy radius]  
user@host# set network-element ne2
```

Configuring the Authentication Target Used by the SIC Server (SRC CLI)

The authentication target is a defined network element target that can then be assigned to an authentication route. It refers to a downstream RADIUS AAA server. You must specify a previously configured downstream RADIUS network element name.

Use the following statements to configure the authentication target for the SIC group:

```
shared sic group identifier server identifier authentication-route id target {
network-element network-element;
```

To configure the authentication route target used by the SIC server:

1. From configuration mode, access the statement that configures the authentication route target. For example, to configure an authentication route called aaa-route2 for the SIC group group2 and server svr2:

```
[edit]
user@host# edit shared sic group group2 server svr2 authentication-route aaa-route2
```

2. Specify the name of the previously configured downstream network element you want to use as the authentication target. For example, to configure the preconfigured network element called ne2 as the authentication target:

```
[edit shared sic group group2 server svr2 authentication-route aaa-route2]
user@host# set network-element ne2
```

Related Documentation

- [Accounting Method and Target \(SRC CLI\) on page 13](#)
- [Request Routing \(SRC CLI\) on page 14](#)

Configuring Request Routing (SRC CLI)

- [Configuration Statements for SIC Explicit Accounting Routing Rules on page 83](#)
- [Configuration Statements for SIC Explicit Authentication Routing Rules on page 84](#)
- [Configuring Explicit Routing \(SRC CLI\) on page 85](#)
- [Configuring Implicit Routing \(SRC CLI\) on page 88](#)

Configuration Statements for SIC Explicit Accounting Routing Rules

Use the following statements to configure explicit routing rules for the SIC at the **[edit]** hierarchy level:

```
shared sic group identifier server identifier accounting-route
shared sic group identifier server identifier accounting-route id condition realm {
  (present | not-present);
}
shared sic group identifier server identifier accounting-route id condition realm
  does-not-equal value
shared sic group identifier server identifier accounting-route id condition realm equals value
shared sic group identifier server identifier accounting-route id condition realm has-prefix
  value
shared sic group identifier server identifier accounting-route id condition realm has-suffix
  value
shared sic group identifier server identifier accounting-route id condition realm range {
  low low;
  high high;
```

```

}
shared sic group identifier server identifier accounting-route id condition request
shared sic group identifier server identifier accounting-route id condition request attribute
  attribute-name {
    (present | not-present);
  }
shared sic group identifier server identifier accounting-route id condition request attribute
  attribute-name does-not-equal value
shared sic group identifier server identifier accounting-route id condition request attribute
  attribute-name equals value
shared sic group identifier server identifier accounting-route id condition request attribute
  attribute-name has-prefix value
shared sic group identifier server identifier accounting-route id condition request attribute
  attribute-name has-suffix value
shared sic group identifier server identifier accounting-route id condition request attribute
  attribute-name range {
    low low;
    high high;
  }
shared sic group identifier server identifier accounting-route id condition user-identity {
  (present | not-present);
}
shared sic group identifier server identifier accounting-route id condition user-identity
  does-not-equal value
shared sic group identifier server identifier accounting-route id condition user-identity equals
  value
shared sic group identifier server identifier accounting-route id condition user-identity
  has-prefix value
shared sic group identifier server identifier accounting-route id condition user-identity
  has-suffix value
shared sic group identifier server identifier accounting-route id condition user-identity range
  {
    low low;
    high high;
  }

```

Configuration Statements for SIC Explicit Authentication Routing Rules

Use the following statements to configure explicit routing rules for the SIC at the **[edit]** hierarchy level:

```

shared sic group identifier server identifier authentication-route id
shared sic group identifier server identifier authentication-route id editing editing-rule
shared sic group identifier server identifier authentication-route id target {
}
shared sic group identifier server identifier authentication-route id condition user-identity
  {
    (present | not-present);
  }
shared sic group identifier server identifier authentication-route id condition user-identity
  range {
    low low;
    high high;
  }
shared sic group identifier server identifier authentication-route id condition user-identity
  equals value

```

```

shared sic group identifier server identifier authentication-route id condition user-identity
  does-not-equal value
shared sic group identifier server identifier authentication-route id condition user-identity
  has-prefix value
shared sic group identifier server identifier authentication-route id condition user-identity
  has-suffix value
shared sic group identifier server identifier authentication-route id condition realm {
  (present | not-present);
}
shared sic group identifier server identifier authentication-route id condition realm range {
  low low;
  high high;
}
shared sic group identifier server identifier authentication-route id condition realm equals
  value
shared sic group identifier server identifier authentication-route id condition realm
  does-not-equal value
shared sic group identifier server identifier authentication-route id condition realm has-prefix
  value
shared sic group identifier server identifier authentication-route id condition realm has-suffix
  value
shared sic group identifier server identifier authentication-route id condition request {
}
shared sic group identifier server identifier authentication-route id condition request attribute
  attribute-name {
  (present | not-present);
}
shared sic group identifier server identifier authentication-route id condition request attribute
  attribute-name range {
  low low;
  high high;
}
shared sic group identifier server identifier authentication-route id condition request attribute
  attribute-name equals value
shared sic group identifier server identifier authentication-route id condition request attribute
  attribute-name does-not-equal value
shared sic group identifier server identifier authentication-route id condition request attribute
  attribute-name has-prefix value
shared sic group identifier server identifier authentication-route id condition request attribute
  attribute-name has-suffix value

```

Configuring Explicit Routing (SRC CLI)

Explicit routing rules can be configured for accounting and authentication requests. When you configure an accounting or authentication route, you specify:

- (Optional) An editing rule you want to apply to the request before it is forwarded to the target.
- A predefined accounting method that is the target for the route.
- A predefined authentication route target (network element).
- (Optional) A set of conditions that must be matched in the request for the route to be selected.

When multiple routes are configured, they are evaluated in the order they are displayed by the **show** command. A newly created route is displayed last among the routes and has the lowest priority, so it is evaluated last. You can use the SRC CLI **insert** command to move a route before or after another route to change its evaluation order. The higher a route is displayed on the list, the sooner it is evaluated.

You can specify any combination of match conditions and condition tests as described in [Table 11 on page 86](#). For a complete list of statements used to configure explicit routing rules, see “[Configuration Statements for SIC Explicit Accounting Routing Rules](#)” on page 83 and “[Configuration Statements for SIC Explicit Authentication Routing Rules](#)” on page 84.

Table 11: Explicit Routing Rule Conditions

Match Condition	Condition Tests
<ul style="list-style-type: none"> • Realm • User identity • Request attribute 	<ul style="list-style-type: none"> • Present • Not present • Equals • Does not equal • Has suffix • Has prefix • Within range

For a complete list of statements you use to configure accounting routes, see “[Configuration Statements for SIC Explicit Accounting Routing Rules](#)” on page 83. For a complete list of statements you use to configure authentication routes, see “[Configuration Statements for SIC Explicit Authentication Routing Rules](#)” on page 84.

To configure explicit routes:

1. From configuration mode, access the configuration statement used to configure explicit routes. For example, to configure an accounting route called route66 for the server svr1, in a group called g1:

```
[edit]
user@host# edit shared sic group g1 server svr1 accounting-route route66
```

2. (Optional) Specify the name of the predefined editing rule you want applied to the request before it is forwarded to the target. For example to apply an editing rule called er1:

```
[edit shared sic group g1 server svr1 accounting-route route66]
user@host# edit editing er1
```

3. Specify a predefined accounting method or authentication routing target to use as the target of the route. If this route is selected, packets are routed to this target. For example, to specify an accounting method called acctg-meth1 as the target:

```
[edit shared sic group g1 server svr1 accounting-route route66 editing er1]
```



```

user@host# up
[edit shared sic group g1 server svr1 accounting-route route66 editing]
user@host# up
[edit shared sic group g1 server svr1 accounting-route route66]
user@host# edit target
[edit shared sic group g1 server svr1 accounting-route route66 target]
user@host# set accounting-method acctg-meth1

```

4. (Optional) Specify the conditions that must be matched for the route to be selected. For example, to specify that the request must contain a realm=abc.com:

```

[edit shared sic group g1 server svr1 accounting-route route66 target]
user@host# up
[edit shared sic group g1 server svr1 accounting-route route66]
user@host# edit condition realm equals abc.com
[edit shared sic group g1 server svr1 accounting-route route66 condition realm equals
  abc.com]
user@host#

```

5. Commit the configuration.

```

user@host# commit
commit complete.

```

6. Verify the routing configuration.

```

[edit shared sic group g1 server svr1 accounting-route route66 condition realm equals
  abc.com]
user@host# up
[edit shared sic group g1 server svr1 accounting-route route66 condition realm]
user@host# up
[edit shared sic group g1 server svr1 accounting-route route66 condition]
user@host# up
[edit shared sic group g1 server svr1 accounting-route route66]
user@host# show

```

```

condition {
  realm {
    equals abc.com;
  }
}
editing {
  er1;
}
target {
  accounting-method acctg-meth1;
}

```

```

[edit shared sic group g1 server svr1 accounting-route route66]
user@host#

```

Configuring Implicit Routing (SRC CLI)

You configure implicit accounting and authentication routes by specifying the name of a previously configured network element that has the proxy function assigned to it. You can also define a default route used for all requests from all realms, or you can specify that only requests from specific realms are routed to the proxy AAA server. When you specify specific realms, you have the option to set a condition of either an exact match of the realm string, or a match on the prefix of the realm string.

Use the following statements to configure implicit routes for the SIC:

```
shared sic group identifier radius network-element id proxy {  
}  
shared sic group identifier radius network-element id proxy realm realmValue {  
  condition (exact | prefix);  
}
```

To configure implicit routes for the SIC:

1. From configuration mode, access the statement that configures the remote AAA server as a proxy. For example, to configure the AAA server in a network element called `ne1` as a proxy:

```
[edit]  
user@host# edit shared sic group group1 radius network-element ne1 proxy
```

2. (Optional) Specify that only requests from specific realms are routed to the proxy AAA server by specifying the names of the realms. For example, to specify that all requests from the realm called `abc.com` are routed to the proxy AAA server:

```
[edit shared sic group group1 radius network-element ne1 proxy]  
user@host# edit realms abc.com
```

3. (Optional) Specify the match condition for the realm.

```
[edit shared sic group group1 radius network-element ne1 proxy realm abc.com]  
user@host# set condition exact | prefix
```

4. (Optional) Specify whether you want this proxy AAA server to be the default route for requests from all realms.

```
[edit shared sic group group1 radius network-element ne1 proxy]  
user@host# set default-route-for-all-realms
```

Configuring Event Logging for an SIC Server (SRC CLI)

You can configure the SIC server to capture any number of log streams called loggers. If you configure multiple log streams, make sure you configure unique names for each log stream. You can configure the log stream to display only log messages from particular log groups. To configure the event level for a log group, you first specify the log group and then specify the event level for it.

Use the following statements to configure event logging for the SIC server:

```
shared sic group identifier server identifier

shared sic group identifier server identifier logger id

shared sic group identifier server identifier logger id file {
  filter (/error | /debug-error);
  filename filename;
  maximum-file-size maximum-file-size;
  rollover-interval rollover-interval;
  rollover-on-startup;
  flush-after-writes;
  high-resolution-timestamps;
  header header;
  footer footer;
  prepend-message-header;
  work-id-label work-id-label;
  work-id-padding work-id-padding;
  utc;
}
shared sic group identifier server identifier logger id group (administration | configuration
| system | packet | packet-trace | packet-trace-raw) {
  events (error | warning | standard | detail | debug);
}
```

To configure event logging for the SIC server:

1. From configuration mode, access the statement that configures the server belonging to the SIC group. For example, to configure the server called `sicsr1` for the group `group1`:

```
[edit]
user@host# edit shared sic group group1 server sicsr1
```

2. Specify the name used by the server to identify the log stream.

```
[edit shared sic group group1 server sicsr1 logger]
user@host# set id log1
```

3. (Optional) Specify the filter to define which event messages are logged or ignored.

```
[edit shared sic group group1 server sicsr1 logger log1 file]
user@host# set filter (/error | /debug-error)
```

where:

- `/error`—Error events are captured for every log group
 - `/debug-error`—Debug events are captured for every log group
4. Specify the prefix to be added to the log file for easy identification.

```
[edit shared sic group group1 server sicsr1 logger log1 file]
user@host# set filename filename
```

5. (Optional) Specify the maximum size of the log file and the rollover file.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set maximum-file-size maximum-file-size
```



NOTE: The maximum file size is specified in KB. Maximum size of the log file is 10,000,000 KB.

Do not set the maximum file size to a value greater than the available disk space.

6. (Optional) Specify the time in seconds for the rollover interval after which the new log file is created.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set rollover-interval rollover-interval
```

7. (Optional) Specify whether the new log file is to be created every time the server starts.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set rollover-on-startup
```

8. (Optional) Specify whether or not to buffer log messages.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set flush-after-writes
```

- If set, log messages are immediately written to the log file without buffering. Use this setting for real-time logging.
- If not set, SIC log messages are kept in the buffer until the buffer is full and then all messages in the buffer are written to the log file. Use this setting for performance optimization, when real-time logging is not needed.

9. (Optional) Specify whether the high-resolution-time reporting system functions are used.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set high-resolution-timestamps
```

10. (Optional) Specify the header message to be added to the beginning of each log file.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set header header
```

11. (Optional) Specify the footer message to be added to the end of each log file.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set footer footer
```

12. (Optional) Specify whether to prepend each log message with additional information such as time, thread, and transaction information.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set prepend-message-header
```

13. (Optional) Specify the work data ID prefix to be added to each log message.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set work-id-label work-id-label
```

14. (Optional) Specify the string to be added to each log message if work data is not available.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set work-id-padding work-id-padding
```

15. (Optional) Specify the time and date values to Universal Time Coordinates (UTC, formerly known as Greenwich Mean Time, or GMT). If disabled, time and date reflect local time.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set utc
```

16. Configure the event level for each log group for which you want to collect events. First, specify the name of the log group, and then specify the event level. Repeat the process for each log group for which you want to collect events.

```
[edit]
user@host# edit shared sic group group1 server sicser1 logger log1 group (administration
| configuration | system | packet | packet-trace | packet-trace-raw)
```

Where:

- **administration**—Log group reports events related to server administration.
- **configuration**—Log group reports events related to server configuration.
- **system**—Log group reports events related to the system, such as system start and system stop.
- **packet**—Log group reports events related to transaction processing, such as incoming and outgoing packets.
- **packet-trace**—Log group displays contents of a packet. The format is attribute name:attribute value.
- **packet-trace-raw**—Log group displays raw data (octets) of incoming and outgoing packets.

17. (Optional) Specify the highest event level for the log group.

```
[edit shared sic group group1 server sicser1 logger log1 group]
user@host# set events (error | warning | standard | detail | debug)
```

Where:

- **error**—Messages in log shown at event level error.
- **warning**—Messages in log shown at event levels error and warning.
- **standard**—Messages in log shown at event levels error, warning, and standard.
- **detail**—Messages in log shown at event levels error, warning, standard, and detail.
- **debug**—Messages in log shown at event levels error, warning, standard, detail, and debug.

**Related
Documentation**

- [SIC Event Logging Overview \(SRC CLI\) on page 29](#)
- [Configuring SNMP for the SIC Group \(SRC CLI\) on page 92](#)
- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 12](#)
- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 155](#)

Configuring SNMP for the SIC Group (SRC CLI)

You can configure each SNMP event and associated dilution factor. When an event occurs, an SNMP trap is sent to the SNMP manager.

Use the following statements to configure SNMP for the SIC server:

```
shared sic group identifier snmp event ( sic-server-startup | sic-server-shutdown |
    sic-server-unauthorized-administration-request | sic-server-internal-error |
    sic-server-resource-failure | sic-server-log-file-failure ) {
    dilution-factor dilution-factor;
}
```

To configure SNMP events for the SIC group:

1. Specify the SNMP trap name for which you want to configure the dilution factor.

```
[edit]
user@host# edit shared sic group group1 snmp event ( sic-server-startup |
    sic-server-shutdown | sic-server-unauthorized-administration-request |
    sic-server-internal-error | sic-server-resource-failure | sic-server-log-file-failure )
```

Where:

- **sic-server-startup**—SNMP trap on server startup.
- **sic-server-shutdown**—SNMP trap on server shutdown.
- **sic-server-unauthorized-administration-request**—SNMP trap on unauthorized administration request.
- **sic-server-internal-error**—SNMP trap on server internal error.
- **sic-server-resource-failure**—SNMP trap on server resource failure.
- **sic-server-log-file-failure**—SNMP trap on server log file failure.

2. (Optional) Specify the dilution factor. The event is sent to the SNMP manager every *n* occurrences of the condition that generated the alert.

[edit shared sic group group1 snmp event]

user@host# **set dilution-factor *dilution-factor***

**Related
Documentation**

- [SNMP Support for the SIC Overview \(SRC CLI\) on page 32](#)
- [Configuring Event Logging for an SIC Server \(SRC CLI\) on page 88](#)

CHAPTER 5

Configuration Tasks for RADIUS COA

- [Device and Service Template Configuration Overview \(SRC CLI\) on page 95](#)
- [SIC RADIUS Dynamic Authorization Configuration Summary \(SRC CLI\) on page 101](#)
- [Configuring Device Templates \(SRC CLI\) on page 102](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 103](#)
- [Configuration Statements for SIC Service Templates \(SRC CLI\) on page 104](#)
- [Configuring SIC Service Templates \(SRC CLI\) on page 105](#)
- [Configuration Statements for Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 113](#)
- [Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 114](#)
- [Configuration Statements for SIC Global Service Templates \(SRC CLI\) on page 121](#)
- [Configuring Global Service Templates \(SRC CLI\) on page 122](#)
- [Configuring Management of RADIUS-Enabled Devices for the SIC \(SRC CLI\) on page 130](#)
- [Configuring Upstream Network Elements and Dynamic Authorization Targets \(SRC CLI\) on page 131](#)
- [SIC Diameter Configuration Summary \(SRC CLI\) on page 132](#)
- [Configuring the SIC Diameter Server \(SRC CLI\) on page 133](#)
- [Configuring the Diameter Application \(SRC CLI\) on page 137](#)
- [Configuring Diameter Peers \(SRC CLI\) on page 143](#)
- [Configuring the NAS Groups \(SRC CLI\) on page 145](#)
- [Configuring the SAE to Manage AAA Devices on page 150](#)
- [Configuring AAA Policies \(SRC CLI\) on page 152](#)

Device and Service Template Configuration Overview (SRC CLI)

To configure dynamic authorization using the SIC you need to configure:

- **Device template**—Specifies the router make, model and capability.
- **Service template**—Specifies any services that you want to enable for your router. What services are available vary from router to router, so it is important that you understand the properties of your router to successfully implement custom services.

- **Global service template**—Specifies rendering used as part of any mode of any service template. Global service templates are used to control rendering of service-independent requests, such as Abort-Session. A global service template is unique in that its modes, attributes, and variables are available to all services that you define. Global service templates are therefore a mandatory part of any SIC COA configuration.

Device Template Configuration Overview (SRC CLI)

Device templates specify the activation behavior of services and how the router handles multiple requests.

To configure device templates, you specify the capability and its associated value. The associated value is dependent on the specified capability. [Table 12 on page 96](#) describes the available capabilities and associated values.

Table 12: Device Template Capabilities and Associated Values

Capability	Value
Activation —Specify service access/activation behavior.	None (default value)—Indicates that the router is not capable of activating services during initial authorization or activation.
	Access-Accept —Indicates that the router supports activating services only in RADIUS Access-Accept messages.
	CoA —Indicates that the router supports activating services in COA only.
	Both —Enables both Access-Accept and COA requests.
Modification —Specify service modification behavior.	False (default value)—This attribute must be set to false.
Bundle —Indicates whether and how the router handles multiple service activation/deactivations in one COA.	None (default value)—Indicates no bundling.
	Single —Indicates the router accepts multiple requests.

Service and Global Service Template Configuration Overview (SRC CLI)

Service templates specify any services that you want to enable for your router. What services are available vary from router to router, so it is important that you understand the properties of your router to successfully implement custom services.

Global service templates specify rendering used as part of any mode of any service template. Global service templates are used to control rendering of service-independent requests, such as Abort-Session. A global service template is unique in that its modes, attributes, and variables are available to all services that you define. Global service templates are therefore a mandatory part of any SIC COA configuration.

You need to configure the following items for both the service and global service template:

- Mode

- Attributes
- Variable

Mode

Service and global service templates have groups of data called mode that each service must specify. A mode contains attributes and variables, which are explained in the next sections. It is mandatory to configure the mode for each service and global template. You must use the provided modes; you cannot create new modes.

Table 13 on page 97 lists the modes and attributes for global service templates.

Table 13: Service Template Modes

Mode	Description
Activation	Activates services on request from the SAE.
Deactivation	Deactivates services on request from the SAE.
Initial-Authorization	Initial activation of services in the Access-Accept message.
Service-Correlation-Id	Assigns an ID number when any other mode is initiated. The SRC software uses this identification number internally.
Service-Profile-Download	Used for Cisco routers only. See “ Caveat (Cisco Only) ” on page 97.

Table 14 on page 97 lists the modes and attributes for global service templates.

Table 14: Global Service Template Modes

Mode	Description
Authentication	<p>Use this mode for optional rendering of the request in the case of an Initial-Authorization. Usually this mode is empty, since no additional rendering is required.</p> <p>Unlike modes in service templates, this mode renders requests to the SRC software and not to the router.</p>
Accounting	Use this mode to control the rendering of the accounting request sent to the SAE. Accounting is a post-authorization service, and it uses the ID numbers and names from the service activation rendering.
Abort-Session	Use this mode for rendering of RADIUS disconnect request (DM) upon abort session request from the SAE.

Caveat (Cisco Only)

Cisco routers require an additional step to complete service activation. When the SIC activates a service on a Cisco router, the router sends an extra Access-Request to the SIC to retrieve the service profile. The SIC then sends back an Access-Accept response with VSAs representing the service profile. In response to the extra Access-Request, the

SIC has to send all VSAs generated by the previous rendering process. The router then activates the service. This means that the SIC has to render the activation twice. In the second rendering a special mode, Service-Profile-Download is used.

This activation process is different from the usual scenario. Extra Access-Requests happen prior to the SIC response to an SAE request. Therefore, you can minimize the first rendering and place most of the work on the SAE download mode by doing the following:

The **Service-Profile-Download** mode in the supplied Cisco router configuration template is used to render the answer to the Cisco Profile Download request. The **Initial-Authorization** or **Activation** modes are used to render the first Access-Accept or COA message in the packet. To comply with the Cisco requirement to have only the service name in the first Access-Accept or COA message, the **Initial-Authorization** or **Activation** modes should contain the attribute for the service name only, and the rest of parameters should be specified using the **shared sic group identifier device-template id service-template name mode service-profile-download** statement.

- In the activation mode, specify only the service name.
- In the service-policy-download mode, specify the rest of the needed parameters.

See your Cisco documentation for more information.

Attributes

All modes have attributes. Attributes define which RADIUS attributes are generated as a result of rendering. All attributes create data that appears in the RADIUS attributes (such as VSAs) generated by the rendering process. It is important to understand that modes are the very core of the rendering process.

Table 15 on page 98 lists the attributes, explains their parameters, and describes their behavior.

Table 15: Attributes for All Modes

Attribute	Description
required	<p>If the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message for the router. Otherwise, the rendering fails.</p> <p>Options</p> <ul style="list-style-type: none"> • name name—Name of the attribute. The specified name must match a defined RADIUS attribute in the downstream AAA server response. • copy-from copy-from—(Optional) Specify the name of the attribute to copy the value from. If the copy-from option is specified, the renderer looks up the attribute specified by copy-from option in the downstream AAA Server response. In the absence of copy-from option, the renderer looks up the attribute specified by the name option.

Table 15: Attributes for All Modes (*continued*)

Attribute	Description
override	<p>Whether or not the renderer finds the attribute in the downstream AAA server response, it creates the attribute name with the specified value.</p> <p>Options</p> <ul style="list-style-type: none"> • name <i>name</i>—Name of the attribute. The name must match a defined RADIUS attribute in the downstream AAA server response. • value <i>value</i>—Set the attribute to this value.
default	<p>If the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message. Otherwise, it creates the attribute name with the specified value.</p> <p>Options</p> <ul style="list-style-type: none"> • name <i>name</i>—Name of the attribute. The name must match a defined RADIUS attribute in the downstream AAA server response. • value <i>value</i>—Set the attribute to this value. • copy-from <i>copy-from</i>—(Optional) Specify the name of the attribute to copy the value from. If the copy-from option is specified, the renderer looks up the attribute specified by copy-from option in the downstream AAA Server response. In the absence of copy-from option, the renderer looks up the attribute specified by the name option.
normal	<p>If the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message for the router. Otherwise, no action occurs. Unlike <i>required-attribute</i>, the rendering does not fail in this case.</p> <p>Options</p> <ul style="list-style-type: none"> • name <i>name</i>—Name of the attribute. The specified name must match a defined RADIUS attribute in the downstream AAA server response. • copy-from <i>copy-from</i>—(Optional) Specify the name of the attribute to copy the value from. If the copy-from option is specified, the renderer looks up the attribute specified by copy-from option in the downstream AAA Server response. In the absence of copy-from option, the renderer looks up the attribute specified by the name option.

Table 15: Attributes for All Modes (*continued*)

Attribute	Description
parameterized	<p>The most powerful and flexible part of the template. It generates attribute values using a format specification, which makes it the most flexible of the attributes.</p> <p>Options</p> <ul style="list-style-type: none"> • name <i>name</i>—Name of the attribute. The specified name must match a defined RADIUS attribute in the downstream AAA server response. • format <i>format</i>—In a form of "\$ (p1) \$ (p2) ... \$ (pn) [\$p(n+1)]". Behaves much like <code>sprintf</code> in C; you can intersperse literal text in between parameter definitions. Unlike <code>sprintf</code>, <code>format</code> supports an optional parameter definition. If the optional parameter is absent, it, and any literal text included in the square brackets, is ignored. All parameters come from the SAE as input to rendering. If you need to use restricted characters in your strings, use the backslash convention: <code>\\$, \', \", \[, \], \(\, \)</code>.

Variables

Modes can also have variables, which control the rendering process. Variables are subtags under modes. You can use them to render information that is not part of RADIUS attributes. They provide inner logic for the rendering process. Nothing defined by variables appears in VSAs sent to the router.

Variables have three configuration options, described in [Table 16 on page 100](#).

Table 16: Variables

Option	Description
name	The variable name
value	The value, usually an integer
type	The data type, integer or string

A rule for processing variables: while rendering, when the SIC encounters a variable with a new value, and that variable already has a different value, the rendering stops and sends the results to the SAE. The SAE generates a RADIUS message and resumes rendering with the new value. Thus, it creates two VSAs, one each for the variable values. This correlates with the Bundle capability.

Overriding the Service Correlation ID

You can also use variables to override the **service-correlation-id** mode. For example,

```
variable name= "CreateServiceCorrelationId" value="0"
```

overrides the **service-correlation-id** mode, so no identification number is created.

Tagged Attributes

The SIC supports tagged attributes, which are an extension of the RADIUS protocol. Refer to RFC 2868 (<http://www.ietf.org/rfc/rfc2868>) for a description of this feature.

If you have **bundle=single** and you want to send a single COA activating two services, these activation requests must have the same RADIUS attributes, but with different values. To discriminate between attributes from two separate activation requests, you must use a unique tag for each.

Specify tagged attributes using the **shared sic group *identifier* device-template *id* service-template *name* mode** (**activation|deactivation|initial-authorization|service-correlation-id|service-profile-download**) **attributes tagged-group *name*** statement.



NOTE: Each service template is restricted to have only one tagged group; for attributes configured under the tagged-group, only attributes that support tags are affected. Otherwise, it has no effect if the configured attributes does not support tagging.

The attributes described in [Table 15 on page 98](#) are also support for tagged attribute configurations.

Related Documentation

- [Managing Dynamic Services on RADIUS-Enabled Devices on page 35](#)
- [SIC Dynamic Authorization Support Overview on page 36](#)
- [How the Dynamic Authorization Process Works in the SIC on page 38](#)

SIC RADIUS Dynamic Authorization Configuration Summary (SRC CLI)

To configure RADIUS dynamic authorization support, you must configure the device and service templates:

- Review the device and service template configuration overview.
See [“Device and Service Template Configuration Overview \(SRC CLI\)” on page 95](#).
- Configure the device template used by the SIC group.
See [“Configuring Device Templates \(SRC CLI\)” on page 102](#).
- Configure the device capabilities.
See [“Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\)” on page 103](#).
- Configure the service template.
See [“Configuring SIC Service Templates \(SRC CLI\)” on page 105](#).

- Configure any tagged attributes for the service template.
See [“Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\)”](#) on page 114.
- Configure the global service template.
See [“Configuring Global Service Templates \(SRC CLI\)”](#) on page 122.

**Related
Documentation**

- [Managing Dynamic Services on RADIUS-Enabled Devices](#) on page 35
- [SIC Dynamic Authorization Support Overview](#) on page 36
- [How the Dynamic Authorization Process Works in the SIC](#) on page 38

Configuring Device Templates (SRC CLI)

Device templates specify the make (vendor), model, and capability of the router. Device models are stored in the Juniper Networks database and can be shared by multiple SICs.



NOTE: When you modify a device template, you must restart the SIC to apply the changes.

Before you configure the device template, you need to configure the device models and dictionaries used by the SIC group. See [“Configuring the Device Models Supported by the SIC Group \(SRC CLI\)”](#) on page 54 and [“Configuring Dictionaries for the SIC Group \(SRC CLI\)”](#) on page 51.

Use the following statements to configure a device template for the SIC:

```
shared sic group identifier device-template id {  
    vendor vendor;  
    model model;  
}
```

To configure a device template for the SIC:

1. From configuration mode, access the statement that configures the device template and specify a name for the template. For example, to create a device template named `dt1` in an SIC group named `g1`:

```
[edit]  
user@host# edit shared sic group g1 device-template dt1
```

We provide templates for Juniper Networks E Series Broadband Services Routers running JunosE Software release 7.2 or later and for Cisco routers running Cisco IOS Release 12.2SB. These templates include sample global and service templates that you can modify for your specific environment. To specify the Juniper Networks or Cisco template, enter the following device template names:

- `juniper-router-junose-7.2-plus`
- `cisco-router-ios-12.2-sb`

2. (Optional) Specify the vendor supported in the device template.

```
[edit shared sic group g1 device-template dt1]
user@host# set vendor vendor
```

3. (Optional) Specify the device model name supported in the device template.

```
[edit shared sic group g1 device-template dt1]
user@host# set model model
```

Related Documentation

- [SIC Dictionaries and Device Models Overview \(SRC CLI\) on page 28](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 103](#)
- [Device and Service Template Configuration Overview \(SRC CLI\) on page 95](#)
- [Sample Service Templates on page 165](#)

Configuring the Device Capabilities Supported in the Device Template (SRC CLI)

Device capabilities specify access behavior, modification of the existing service, and whether multiple COAs can attach to one VSA.

Use the following statements to configure the device capabilities:

```
shared sic group identifier device-template id capabilities capability (activation |
  modification | bundle) {
  value;
}
```

To configure device capabilities, you specify the capability and its associated value. The associated value is dependent on the specified capability. [Table 17 on page 103](#) describes the available capabilities and associated values.

Table 17: Capabilities and Associated Values

Capability	Value
Activation —Specify service access or activation behavior.	None (default value)—Indicates that the router is not capable of activating services during initial authorization or activation.
	Access-Accept —Indicates that the router supports activating services only in RADIUS Access-Accept messages.
	COA —Indicates that the router supports activating services in COA only.
	Both —Enables both Access-Accept and COA requests.
Modification —Specify service modification behavior.	False (default value)—This attribute must be set to false.

Table 17: Capabilities and Associated Values (*continued*)

Capability	Value
Bundle —Indicates whether and how the router handles multiple service activations or deactivations in one COA.	None (default value)—Indicates no bundling. Single —Indicates that the router accepts multiple requests.

To configure the device capabilities:

1. From configuration mode, access the statement that configures the device capabilities and specify the capability you want to configure. The following sample procedure uses `g1` as the SIC group name and `dt1` as the device template name.

```
[edit]
user@host# edit shared sic group identifier device-template id capabilities capability
(activation | modification | bundle)
```

For example, to specify the **bundle** capability with a value of **single**, enter:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 capabilities capability bundle
```

2. Specify a value for the capability. For example, to set the **bundle** capability to the value of **single**:

```
[edit shared sic group g1 device-template dt1 capabilities capability bundle]
user@host# set single
```

Related Documentation

- [Configuring Device Templates \(SRC CLI\) on page 102](#)
- [Device and Service Template Configuration Overview \(SRC CLI\) on page 95](#)
- [Sample Service Templates on page 165](#)

Configuration Statements for SIC Service Templates (SRC CLI)

Use the following statements to configure service templates:

```
shared sic group identifier device-template id service-template name {
  description description;
}
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  variable name {
    value value;
    type (integer | string);
  }
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes {
```

```

}
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes attribute id
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes attribute id required {
    name name;
    copy-from copy-from;
  }
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes attribute id normal {
    name name;
    copy-from copy-from;
  }
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes attribute id default {
    name name;
    value value;
    copy-from copy-from;
  }
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes attribute id parameterized {
    format format;
    name name;
  }
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes attribute id override {
    name name;
    value value;
  }
}

```

Related Documentation

- [Device and Service Template Configuration Overview \(SRC CLI\) on page 95](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 103](#)
- [Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 114](#)
- [Configuring Global Service Templates \(SRC CLI\) on page 122](#)

Configuring SIC Service Templates (SRC CLI)

Service templates are used to specify any services that you want to enable for your router. What services are available vary from router to router, so it is important that you understand the properties of your router to successfully implement custom services.

When you configure a service template, you need to specify the mode, and any variables or attributes you want included in the template.

Refer to “[Device and Service Template Configuration Overview \(SRC CLI\)](#)” on page 95 for details on configuring the options in the following procedure.

- [Creating an SIC Service Template \(SRC CLI\)](#) on page 106
- [Configuring the Mode of the SIC Service Template \(SRC CLI\)](#) on page 106
- [Configuring Variables for the SIC Service Template \(SRC CLI\)](#) on page 107
- [Configuring Normal Attributes for the SIC Service Template \(SRC CLI\)](#) on page 107
- [Configuring Required Attributes for the SIC Service Template \(SRC CLI\)](#) on page 109
- [Configuring Default Attributes for the SIC Service Template \(SRC CLI\)](#) on page 110
- [Configuring Parameterized Attributes for the SIC Service Template \(SRC CLI\)](#) on page 111
- [Configuring Override Attributes for the SIC Service Template \(SRC CLI\)](#) on page 112

Creating an SIC Service Template (SRC CLI)

Use the following statements to create an SIC service template:

```
shared sic group identifier device-template id service-template name {  
    description description;  
}
```

To create an SIC service template:

1. From configuration mode, access the statement that configures the service template and specify the name of the template.

```
[edit]  
user@host# edit shared sic group identifier device-template id service-template name
```

For example, to specify a service template called st1:

```
[edit]  
user@host# edit shared sic group g1 device-template dt1 service-template st1
```

2. (Optional) Specify a description for the template.

```
[edit shared sic group g1 device-template dt1 service-template st1]  
user@host# set description description
```

Configuring the Mode of the SIC Service Template (SRC CLI)

Use the following statements to configure the mode of service template:

```
shared sic group identifier device-template id service-template name mode (activation |  
    deactivation | initial-authorization | service-correlation-id | service-profile-download)
```

To configure the mode of the SIC service template:

- From configuration mode, access the statement that configures the service template mode. For example, to specify the **activation** mode:

```
[edit]
```

```
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation
```

Configuring Variables for the SIC Service Template (SRC CLI)

Variables control the behavior of the rendering process.

Use the following statements to configure service template variables:

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
variable name {
  value value;
  type (integer | string);
}
```

To configure variables in the service template:

1. From configuration mode, access the statement that configures variables for the service template and specify a name for the variable. For example, to create a variable named `var1`:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation variable var1
```

Specify the type of variable you want to add to the template. For example, to specify an integer for the variable:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
variable var1]
user@host# set type integer
```

Where the type is either:

- integer
- string

2. Specify the value of the variable. For example, to specify a value of 5 for the variable:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
variable var1]
user@host# set value 5
```

Configuring Normal Attributes for the SIC Service Template (SRC CLI)

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes {
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id
```

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id normal {
  name name;
  copy-from copy-from;
}
```

To configure normal attributes to be included in the service template:

1. (Optional) From configuration mode, access the statement that configures normal attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1
```

2. (Optional) Specify the attribute as a normal attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1]
user@host# edit normal
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 normal]
user@host# set name Unisphere-Service-Timeout
```

4. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 normal]
user@host# set copy-from Session-Timeout
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 normal]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1 normal]
user@host#
```

Configuring Required Attributes for the SIC Service Template (SRC CLI)

With required attributes, if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message for the router, otherwise, rendering fails.

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes {
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id required {
name name;
copy-from copy-from;
}
```

To configure required attributes to be included in the service template:

1. (Optional) From configuration mode, access the statement that configures required attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1
```

2. (Optional) Specify the attribute as a required attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1]
user@host# edit required
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 required]
user@host# set name Unisphere-Service-Timeout
```

4. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 required]
user@host# set copy-from Session-Timeout
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 required]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1 required]
user@host#
```

Configuring Default Attributes for the SIC Service Template (SRC CLI)

With default attributes, if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message. Otherwise, it creates the attribute name with the specified value.

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes {
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id default {
name name;
value value;
copy-from copy-from;
}
```

To configure default attributes to be included in the service template:

1. (Optional) From configuration mode, access the statement that configures default attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1
```

2. (Optional) Specify the attribute as a default attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1]
user@host# edit default
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 default]
user@host# set name Unisphere-Service-Timeout
```


- Specify the value of the attribute. For example, to specify the value of 5:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 default]
user@host# set value 5
```

- (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 default]
user@host# set copy-from Session-Timeout
```

If the rendering process finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message. Otherwise, it creates the attribute name with the specified value.

- Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 default]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
value 5;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1 default]
user@host#
```

Configuring Parameterized Attributes for the SIC Service Template (SRC CLI)

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes {
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id parameterized {
format format;
name name;
}
```

To configure parameterized attributes to be included in the service template:

- (Optional) From configuration mode, access the statement that configures parameterized attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
```

```
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1
```

2. (Optional) Specify the attribute as a parameterized attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1]
user@host# edit parameterized
```

3. Specify the format of the parameterized attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 parameterized]
user@host# set format format
```

4. Specify the name of the attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 parameterized]
user@host# set name name
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 parameterized]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1 parameterized]
user@host#
```

Configuring Override Attributes for the SIC Service Template (SRC CLI)

With override attributes, whether or not the renderer finds the attribute in the downstream AAA server response, it creates the attribute name with the specified value.

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes {
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id override {
name name;
value value;
}
```

To configure override attributes to be included in the service template:

1. (Optional) From configuration mode, access the statement that configures override attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1
```

2. (Optional) Specify the attribute as an override attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1]
user@host# edit override
```

3. Specify the name of the override attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 override]
user@host# set name Unisphere-Service-Timeout
```

4. Specify the value of the attribute. For example, to specify a value of 5:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 override]
user@host# set value 5
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 override]
user@host# show
```

```
name Unisphere-Service-Timeout;
value 5;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1 override]
user@host#
```

Related Documentation

- [Configuring Device Templates \(SRC CLI\) on page 102](#)
- [Device and Service Template Configuration Overview \(SRC CLI\) on page 95](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 103](#)
- [Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 114](#)
- [Configuring Global Service Templates \(SRC CLI\) on page 122](#)

Configuration Statements for Tagged Attributes in SIC Service Templates (SRC CLI)

Use the following statements to configure tagged attributes in a service template:

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group {
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id required {
name name;
copy-from copy-from;
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id normal {
name name;
copy-from copy-from;
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id default {
name name;
value value;
copy-from copy-from;
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id parameterized {
format format;
name name;
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group item id override {
name name;
value value;
}
```

**Related
Documentation**

- [Device and Service Template Configuration Overview \(SRC CLI\) on page 95](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 103](#)
- [Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 114](#)
- [Configuring Global Service Templates \(SRC CLI\) on page 122](#)

Configuring Tagged Attributes in SIC Service Templates (SRC CLI)

The examples in the following procedures use the following:

- sic group=g1
- device template=dt1

- service template=st1
- mode=activation
- attribute identifier=attr1
- attribute=Unisphere-Service-Timeout
- [Creating a Tagged Attribute Group in the SIC Service Template \(SRC CLI\) on page 115](#)
- [Configuring Normal Attributes in a Tagged Attribute Group \(SRC CLI\) on page 115](#)
- [Configuring Default Attributes in a Tagged Attribute Group \(SRC CLI\) on page 116](#)
- [Configuring Required Attributes in a Tagged Attribute Group \(SRC CLI\) on page 118](#)
- [Configuring Override Attributes in a Tagged Attribute Group \(SRC CLI\) on page 119](#)
- [Configuring Parameterized Attributes in a Tagged Attribute Group \(SRC CLI\) on page 120](#)

Creating a Tagged Attribute Group in the SIC Service Template (SRC CLI)

Use the following statements to create a tagged attribute group in the SIC service template:

```
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes tagged-group {
  }
```

To create a tagged attribute group in the SIC service template:

- From configuration mode, access the statement that configures the tagged attribute group in the service template.

```
[edit]
user@host# edit shared sic group identifier device-template id service-template name
  mode (activation | deactivation | initial-authorization | service-correlation-id |
  service-profile-download) attributes tagged-group
```

Attributes defined within tagged attributes will be tagged when included in the renderer result if this attribute supports tagging.

Configuring Normal Attributes in a Tagged Attribute Group (SRC CLI)

With normal attributes, if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message for the router. Otherwise, no action occurs. Unlike required attributes, the rendering does not fail in this case.

```
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes tagged-group attribute id
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes tagged-group attribute id normal {
    name name;
    copy-from copy-from;
  }
```

To configure normal attributes in the tagged attribute group:

1. (Optional) From configuration mode, access the statement that configures normal attributes in the tagged attribute group and specify an identifier for the attribute. For example, to create an identifier named `attr1`:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1
```

2. (Optional) Specify the attribute as a normal attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1]
user@host# edit normal
```

3. Specify the name of the attribute. For example, to specify the attribute `Unisphere-Service-Timeout`:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 normal]
user@host# set name Unisphere-Service-Timeout
```

4. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the `Session-Timeout` attribute contained in the downstream AAA server response, and place it in the `Unisphere-Service-Timeout` attribute:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 normal]
user@host# set copy-from Session-Timeout
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 normal]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1 normal]
user@host#
```

Configuring Default Attributes in a Tagged Attribute Group (SRC CLI)

With default attributes, if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message. Otherwise, it creates the attribute name with the specified value.

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id
```

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id default {
name name;
value value;
copy-from copy-from;
}
```

To configure default attributes in the tagged attribute group:

1. (Optional) From configuration mode, access the statement that configures default attributes in the tagged attribute group and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1
```

2. (Optional) Specify the attribute as a default attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1]
user@host# edit default
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 default]
user@host# set name Unisphere-Service-Timeout
```

4. Specify the value of the attribute. For example, to specify a value of 5:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 default]
user@host# set value 5
```

5. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 default]
user@host# set copy-from Session-Timeout
```

6. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 default]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
value 5;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
```

```
activation attributes tagged-group attribute attr1 default]
user@host#
```

Configuring Required Attributes in a Tagged Attribute Group (SRC CLI)

With required attributes; if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message for the router. otherwise, the renderer fails.

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id required {
name name;
copy-from copy-from;
}
```

To configure required attributes in the tagged attribute group:

1. (Optional) From configuration mode, access the statement that configures required attributes in the tagged attribute group and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1
```

2. (Optional) Specify the attribute as a required attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1]
user@host# edit required
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 required]
user@host# set name Unisphere-Service-Timeout
```

4. Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 required]
user@host# set copy-from Session-Timeout
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 required]
user@host# show
```



```

copy-from Session-Timeout;
name Unisphere-Service-Timeout;

[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1 required]
user@host#

```

Configuring Override Attributes in a Tagged Attribute Group (SRC CLI)

With override attributes, whether or not the renderer finds the attribute in the downstream AAA server response, it creates the attribute name with the specified value.

```

shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id override {
name name;
value value;
}

```

To configure override attributes in the tagged attribute group:

1. (Optional) From configuration mode, access the statement that configures override attributes in the tagged attribute group and specify an identifier for the attribute. For example, to create an identifier named attr1:

```

[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1

```

2. (Optional) Specify the attribute as an override attribute.

```

[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1]
user@host# edit override

```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```

[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 override]
user@host# set name Unisphere-Service-Timeout

```

4. Specify the value of the attribute. For example, to specify a value of 5:

```

[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 override]
user@host# set value 5

```

5. Verify the configuration.

```

[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 override]

```

```
user@host# show
```

```
name Unisphere-Service-Timeout;  
value 5;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode  
activation attributes tagged-group attribute attr1 override]  
user@host#
```

Configuring Parameterized Attributes in a Tagged Attribute Group (SRC CLI)

Parameterized attributes is the most powerful and flexible part of the template. It generates attribute values using a format specification, which makes it the most flexible of the attributes.

```
shared sic group identifier device-template id service-template name mode (activation |  
deactivation | initial-authorization | service-correlation-id | service-profile-download)  
attributes tagged-group name attribute id  
shared sic group identifier device-template id service-template name mode (activation |  
deactivation | initial-authorization | service-correlation-id | service-profile-download)  
attributes tagged-group name attribute id parameterized {  
format format;  
name name;  
}
```

To configure parameterized attributes in the tagged attribute group:

1. (Optional) From configuration mode, access the statement that configures parameterized attributes in the tagged attribute group and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]  
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode  
activation attributes tagged-group attribute attr1
```

2. (Optional) Specify the attribute as a normal attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation  
attributes tagged-group attribute attr1]  
user@host# edit parameterized
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation  
attributes tagged-group attribute attr1 parameterized]  
user@host# set name Unisphere-Service-Timeout
```

4. Specify the format of the parameterized attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation  
attributes tagged-group attribute attr1 parameterized]  
user@host# set format format
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 parameterized]
user@host# show
```

```
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1 parameterized]
user@host#
```

**Related
Documentation**

- [Device and Service Template Configuration Overview \(SRC CLI\) on page 95](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 103](#)
- [Configuring Global Service Templates \(SRC CLI\) on page 122](#)

Configuration Statements for SIC Global Service Templates (SRC CLI)

Use the following statements to configure a global service template:

```
shared sic group identifier device-template id global-template {
  description description;
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session)
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) variable name {
  value value;
  type (integer | string);
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes {
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id required {
  name name;
  copy-from copy-from;
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id normal {
  name name;
  copy-from copy-from;
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id default {
  name name;
  value value;
  copy-from copy-from;
}
```

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id parameterized {
  format format;
  name name;
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id override {
  name name;
  value value;
}
```

**Related
Documentation**

- [Device and Service Template Configuration Overview \(SRC CLI\) on page 95](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 103](#)
- [Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 114](#)
- [Configuring Global Service Templates \(SRC CLI\) on page 122](#)

Configuring Global Service Templates (SRC CLI)

A global service template is a unique service template that specifies rendering used as part of any mode of any other service template. It is used to control rendering of service-independent requests, such as AbortSession. This template is unique in that its modes, attributes, and variables are available to all services that you define. It is therefore a mandatory part of any router configuration. The global service template is called in every possible scenario.

The examples in this procedure use the following configuration:

- sic group=g1
- device template=dt1
- mode=authentication
- attribute identifier=attr1
- attribute=Unisphere-Service-Timeout
- [Creating an SIC Global Service Template \(SRC CLI\) on page 123](#)
- [Configuring the Mode of the SIC Global Service Template \(SRC CLI\) on page 123](#)
- [Configuring Variables for the SIC Global Service Template \(SRC CLI\) on page 123](#)
- [Configuring Normal Attributes for the SIC Global Service Template \(SRC CLI\) on page 124](#)
- [Configuring Required Attributes for the SIC Global Service Template \(SRC CLI\) on page 125](#)
- [Configuring Default Attributes for the SIC Global Service Template \(SRC CLI\) on page 126](#)

- [Configuring Parameterized Attributes for the SIC Global Service Template \(SRC CLI\)](#) on page 128
- [Configuring Override Attributes for the SIC Global Service Template \(SRC CLI\)](#) on page 129

Creating an SIC Global Service Template (SRC CLI)

Use the following statements to create an SIC global service template:

```
shared sic group identifier device-template id global-template {
  description description;
}
```

To create an SIC global service template:

1. From configuration mode, access the statement that configures the global.

```
[edit]
user@host# edit shared sic group identifier device-template id global-template
```

2. (Optional) Specify a description for the template.

```
[edit shared sic group g1 device-template dt1 global-template]
user@host# set description description
```

Configuring the Mode of the SIC Global Service Template (SRC CLI)

Use the following statements to configure the mode of global service template:

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session)
```

To configure the mode of the SIC global service template:

- From configuration mode, access the statement that configures the global service template mode. For example, to specify the **authentication** mode:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 global-template mode
authentication
```

Configuring Variables for the SIC Global Service Template (SRC CLI)

Variables control the behavior of the rendering process.

Use the following statements to configure global service template variables:

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) variable name {
  value value;
  type (integer | string);
}
```

To configure variables in the global service template:

- From configuration mode, access the statement that configures variables for the global service template and specify a name for the variable. For example, to create a variable named `var1`:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 global-template mode
authentication variable var1
```

Specify the type of variable you want to add to the template. For example, to specify an integer for the variable:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
variable var1]
user@host# set type integer
```

Where the type is either:

- integer
- string
- Specify the value of the variable. For example, to specify a value of 5 for the variable:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
variable var1]
user@host# set value 5
```

Configuring Normal Attributes for the SIC Global Service Template (SRC CLI)

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes {
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id normal {
name name;
copy-from copy-from;
}
```

To configure normal attributes to be included in the global service template:

1. (Optional) From configuration mode, access the statement that configures normal attributes and specify an identifier for the attribute. For example, to create an identifier named `attr1`:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1
```

2. (Optional) Specify the attribute as a normal attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1]
user@host# edit normal
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 normal]
user@host# set name Unisphere-Service-Timeout
```

4. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 normal]
user@host# set copy-from Session-Timeout
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 normal]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1 normal]
user@host#
```

Configuring Required Attributes for the SIC Global Service Template (SRC CLI)

With required attributes, if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message for the router, otherwise, rendering fails.

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes {
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id required {
name name;
copy-from copy-from;
}
```

To configure required attributes to be included in the global service template:

1. (Optional) From configuration mode, access the statement that configures required attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1
```

2. (Optional) Specify the attribute as a required attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1]
user@host# edit required
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 required]
user@host# set name Unisphere-Service-Timeout
```

4. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 required]
user@host# set copy-from Session-Timeout
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 required]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1 required]
user@host#
```

Configuring Default Attributes for the SIC Global Service Template (SRC CLI)

With default attributes, if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message. Otherwise, it creates the attribute name with the specified value.

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes {
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id default {
name name;
value value;
copy-from copy-from;
}
```

To configure default attributes to be included in a global service template:

1. (Optional) From configuration mode, access the statement that configures default attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1
```

2. (Optional) Specify the attribute as a default attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1]
user@host# edit default
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 default]
user@host# set name Unisphere-Service-Timeout
```

4. Specify the value of the attribute. For example, to specify the value of 5:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 default]
user@host# set value 5
```

5. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 default]
user@host# set copy-from Session-Timeout
```

If the rendering process finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message. Otherwise, it creates the attribute name with the specified value.

6. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 default]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
value 5;
```

```
[edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1 default]
user@host#
```

Configuring Parameterized Attributes for the SIC Global Service Template (SRC CLI)

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes {
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id parameterized {
format format;
name name;
}
```

To configure parameterized attributes to be included in a global service template:

1. (Optional) From configuration mode, access the statement that configures parameterized attributes and specify an identifier for the attribute. For example, to create an identifier named `attr1`:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1
```

2. (Optional) Specify the attribute as a parameterized attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1]
user@host# edit parameterized
```

3. (Optional) Specify the format of the parameterized attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 parameterized]
user@host# set format format
```

4. Specify the name of the attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 parameterized]
user@host# set name name
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 parameterized]
user@host# show
```

```
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1 parameterized]
user@host#
```

Configuring Override Attributes for the SIC Global Service Template (SRC CLI)

With override attributes, whether or not the renderer finds the attribute in the downstream AAA server response, it creates the attribute name with the specified value.

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes {
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id override {
name name;
value value;
}
```

To configure override attributes to be included in a global service template:

1. (Optional) From configuration mode, access the statement that configures override attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1
```

2. (Optional) Specify the attribute as an override attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1]
user@host# edit override
```

3. Specify the name of the override attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 override]
user@host# set name Unisphere-Service-Timeout
```

4. Specify the value of the attribute. For example, to specify a value of 5:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 override]
user@host# set value 5
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 override]
user@host# show
```

```
name Unisphere-Service-Timeout;
value 5;
```

```
[edit shared sic group g1 device-template dt1 global-template mode
```

```
authentication attributes attribute attr1 override]
user@host#
```

**Related
Documentation**

- [Device and Service Template Configuration Overview \(SRC CLI\) on page 95](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 103](#)
- [Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 114](#)

Configuring Management of RADIUS-Enabled Devices for the SIC (SRC CLI)

To configure management of RADIUS-enabled devices when using the SIC:

1. Configure the NAS group peers, device capabilities, and routes.
[See “Configuring the NAS Groups \(SRC CLI\)” on page 145.](#)
2. Configure the SAE to manage SAE devices.
[See “Configuring the SAE to Manage AAA Devices” on page 150.](#)
3. Configure the AAA policy rules.
[See “Configuring AAA Policies \(SRC CLI\)” on page 152.](#)

**Related
Documentation**

- [Managing Dynamic Services on RADIUS-Enabled Devices on page 35](#)
- [SIC Dynamic Authorization Support Overview on page 36](#)
- [How the Dynamic Authorization Process Works in the SIC on page 38](#)

Configuring Upstream Network Elements and Dynamic Authorization Targets (SRC CLI)

Dynamic authorization targets are logical entities that represent the NAS device in upstream network elements. The SIC forwards COA/DM requests to dynamic authorization targets.

Use the following statements to configure dynamic authorization targets:

```
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  target name {
    address address;
    priority priority;
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  target name {
    secret secret;
    port port;
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  {
    failover-mode (round-robin | primary-backup);
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  failover-policy {
    priority priority;
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  failover-policy retry {
    number number;
    timeout timeout;
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  failover-policy fast-fail {
    minimum-number minimum-number;
    timeout timeout;
    reset-delay reset-delay;
  }
}
```

To configure a dynamic authorization target:

1. From configuration mode, access the statement that configures an upstream network element and dynamic authorization target. For example, to configure an upstream RADIUS network element called `ne1` and dynamic authorization target called `dat1` for the SIC group `group1`:

```
[edit]
user@host# edit shared sic group group1 radius network-element ne1 upstream
dynamic-authorization-target target dat1
```

2. Specify the IP address of the target.

```
[edit shared sic group group1 radius network-element ne1 upstream
dynamic-authorization-target target dat1]
```

```
user@host# set address address
```

3. Specify the priority of the target. Targets with lower priority values are selected before other targets in a failover policy.

```
[edit shared sic group group1 radius network-element ne1 upstream  
dynamic-authorization-target target dat1]  
user@host# set priority priority
```

4. Specify the shared secret used by the target.

```
[edit shared sic group group1 radius network-element ne1 upstream  
dynamic-authorization-target target dat1]  
user@host# set secret secret
```

5. (Optional) Specify the port used by the target to receive dynamic authorization messages.

```
[edit shared sic group group1 radius network-element ne1 upstream  
dynamic-authorization-target target dat1]]  
user@host# set port port
```

**Related
Documentation**

- [SIC Dynamic Authorization Support Overview on page 36](#)
- [RADIUS Authentication/Authorization and Accounting Data Flow on page 8](#)
- [RADIUS and Diameter Configuration for the SIC Overview \(SRC CLI\) on page 21](#)
- [How the Dynamic Authorization Process Works in the SIC on page 38](#)

SIC Diameter Configuration Summary (SRC CLI)

To configure Diameter support for the SIC:

1. Configure the SIC Diameter server including the Diameter network element failover policy and the Diameter peers.

See [“Configuring the SIC Diameter Server \(SRC CLI\)” on page 60](#).

2. Configure the Diameter application.

See [“Configuring the Diameter Application \(SRC CLI\)” on page 137](#).

3. Configure the SRC Diameter server.

See [“Configuring Diameter Peers \(SRC CLI\)” on page 143](#).

**Related
Documentation**

- [RADIUS and Diameter Transports on page 27](#)
- [RADIUS and Diameter Configuration for the SIC Overview \(SRC CLI\) on page 21](#)

Configuring the SIC Diameter Server (SRC CLI)

- [Configuration Statements for the SIC Diameter Server \(SRC CLI\) on page 133](#)
- [Configuring the SIC Diameter Server Identity \(SRC CLI\) on page 134](#)
- [Configuring the SIC Diameter Server Peer \(SRC CLI\) on page 135](#)

Configuration Statements for the SIC Diameter Server (SRC CLI)

Use the following statements to configure the SIC Diameter server:

```
shared sic group identifier server identifier diameter identity {
  origin-host origin-host;
  origin-realm origin-realm;
}
shared sic group identifier server identifier diameter transport id {
  protocol (tcp | sctp);
  port port;
}
shared sic group identifier diameter network-element id {
  description description;
  failover-policy (round-robin | primary-backup);
}
shared sic group identifier diameter network-element id peer name {
  description description;
  address address;
  protocol (tcp | sctp);
  port port;
  active-peer;
  priority priority;
}
shared sic group identifier diameter network-element id peer name {
  enforce-source-address;
}
shared sic group identifier diameter network-element id peer name {
  origin-host origin-host;
}
shared sic group identifier diameter network-element id peer name addresses address
  address
```

Configuring the SIC Diameter Server Identity (SRC CLI)

Configuring the SIC Diameter server identity includes specifying the origin-host, origin-realm, the port the server receives Diameter messages on, and protocol. The SIC Diameter server communicates with the SRC Diameter server. The origin-host and origin-realm identify the SIC Diameter server. This identity is sent in all Diameter requests originating on this server.

The default identity of the SIC Diameter server is set to origin-host="your-host" and the origin-realm="your-realm.net." You must reconfigure these settings for your network environment.

To configure the SRC Diameter server and the Diameter application, see ["Configuring the Diameter Application \(SRC CLI\)" on page 137](#) and ["Configuring Diameter Peers \(SRC CLI\)" on page 143](#).

Use the following statements to configure the SIC Diameter server identity:

```
shared sic group identifier server identifier diameter identity {  
    origin-host origin-host;  
    origin-realm origin-realm;  
}  
shared sic group identifier server identifier diameter transport id {  
    protocol (tcp | sctp);  
    port port;  
}
```

To configure the SIC Diameter server identity:

1. From configuration mode, access the statement that configures the SIC Diameter server. For example, to configure the SIC Diameter server in an SIC group called g1 that includes an SIC server called svr1:

```
[edit]  
user@host# shared sic group g1 server svr1 diameter identity
```

2. Specify the origin-host name of the SIC Diameter server. For example, to specify the origin-host as sic-diam-svr1:

```
[edit shared sic group g1 server svr1 diameter identity]  
user@host# set origin-host sic-diam-svr1
```

3. Specify the origin-realm name of the SIC Diameter server. For example, to specify the origin-realm as abc.com:

```
[edit shared sic group g1 server svr1 diameter identity]  
user@host# set origin-realm abc.com
```

4. Verify your configuration.

```
[edit shared sic group g1 server svr1 diameter identity]  
user@host# show
```



```

user@host# show
origin-host diam-svr1;
origin-realm abc.com;

```

Configuring the SIC Diameter Server Peer (SRC CLI)

The SIC Diameter server handles all communication between the SIC and the SRC Diameter server. This procedure describes how to configure the network element in which the SRC Diameter server logically resides, the failover policy, and the Diameter connection between the SIC Diameter server and the SRC Diameter server.

Use the following statements to configure the SIC Diameter peer:

```

shared sic group identifier diameter network-element id {
  description description;
  failover-policy (round-robin | primary-backup);
}
shared sic group identifier diameter network-element id peer name {
  description description;
  address address;
  protocol (tcp | sctp);
  port port;
  active-peer;
  priority priority;
}
shared sic group identifier diameter network-element id peer name {
  enforce-source-address;
}
shared sic group identifier diameter network-element id peer name {
  origin-host origin-host;
}
shared sic group identifier diameter network-element id peer name addresses address
address

```

To configure the SIC Diameter server peer:

1. From configuration mode, access the statement that configures the SIC Diameter server peer and configure the network element where the SRC Diameter server resides. For example, to configure a Diameter network element called `diam-ne1` for an SIC group called `g1`:

```

[edit]
user@host# shared sic group g1 diameter network-element diam-ne1

```

2. (Optional) Specify a description for the network element.

```

[shared sic group g1 diameter network-element diam-ne1]
user@host# set description description

```

3. (Optional) Configure the failover policy for the network element. For example, to configure the primary or backup failover policy:

```

[shared sic group g1 diameter network-element diam-ne1]
user@host# set primary-backup

```

4. Configure the name of the Diameter peer (SRC Diameter server). For example, to call the peer src-diam-svr1:

```
[shared sic group g1 diameter network-element diam-ne1]
user@host# edit peer src-diam-svr1
```

5. (Optional) Specify a description for the Diameter peer.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set description description
```

6. Specify the IP address of the remote Diameter peer (SRC Diameter server). For example, 10.1.2.3.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set address 10.1.2.3
```

7. Specify the protocol the Diameter peer (SRC Diameter server) uses for Diameter messages (TCP or SCTP).

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set protocol sctp
```

8. Specify which port the Diameter peer (SRC Diameter server) receives messages on. For example, port 2222.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set port 2222
```

9. (Optional) Specify whether the peer is active or not. If the peer is configured to connect actively, the server periodically attempts to connect (or reconnect after a connection has failed) to the remote peer. If this option is not set, a connection is established only after the remote peer attempts to connect to this server.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set active-peer
```

10. (Optional) Specify the priority of the peer for the failover policy. Peers with lower priority values are the preferred routing targets for Diameter requests. Requests are split equally among peers with the same priority level.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set priority 1
```

11. (Optional) Specify whether a source IP match is required for the connection. This option determines whether the source IP address of a connection attempt must match one of the configured IP addresses used to connect to this peer. If this option is not set, requests are accepted from any IP address as long as the client presents the correct host name during the capabilities exchange. This functionality allows other peers to exist behind NAS devices.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set enforce-source-address
```

12. Specify the origin-host name of the Diameter peer (SRC Diameter server). For example, if the origin-host name of the SRC Diameter server is diam-host1:

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set origin-host diam-host1
```

13. (Optional) Specify an ordered set of IP addresses to use for a multilink connection. An IP address of the remote peer is necessary to establish a Diameter connection with the remote peer (SRC Diameter server). For a Diameter connection over TCP, only one configured address is used. Over SCTP, the connection may be established over multiple addresses.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set addresses address 10.1.2.4
user@host# set addresses address 10.1.2.5
user@host# set addresses address 10.1.2.6
```

14. Verify your configuration.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# show
```

```
active-peer;
address 10.1.2.3;
addresses {
  address 10.1.2.4;
  address 10.1.2.5;
  address 10.1.2.6;
}
port 3868;
priority 1;
protocol sctp;
enforce-source-address;
origin-host diam-host1;
```

```
[edit shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host#
```

Configuring the Diameter Application (SRC CLI)

You can configure the properties of the application, client, server, and logging destination of the SRC Diameter application.

Perform the following tasks to configure these properties:

- [Configuring the Diameter Application Properties on page 138](#)
- [Configuring the Diameter Client Properties on page 141](#)
- [Configuring the Diameter Server Properties on page 142](#)
- [Configuring Logging Destinations on page 142](#)

Configuring the Diameter Application Properties

The SRC software supports Diameter application properties such as Juniper Networks Session Resource Control (JSRC) and southbound Gx interface. JSRC and southbound Gx interface communicate with the Service Activation Engine (SAE) (remote SRC peer).

Use the following configuration statements to configure the properties for the Diameter application:

```
system diameter {
  java-heap-size java-heap-size;
  java-new-size java-new-size;
  java-garbage-collection-options java-garbage-collection-options;
  protocol [(tcp | sctp)...];
  local-address [local-address...];
  port port;
  origin-host origin-host;
  origin-realm origin-realm;
  diameter-server-timeout diameter-server-timeout;
  active-peers;
  debug-mode;
  load-balancing-mode (failover | round-robin);
  transaction-processing-log (log-no-messages | log-severe-messages |
    log-normal-messages | log-debug-messages);
  packet-trace-log (log-no-messages | log-severe-messages | log-normal-messages |
    log-debug-messages);
  peer-state-machine-log (log-no-messages | log-severe-messages | log-normal-messages |
    log-debug-messages);
  configuration-log (log-no-messages | log-severe-messages | log-normal-messages |
    log-debug-messages);
}
```

To configure the Diameter application:

1. From configuration mode, access the statement for the Diameter application.

```
user@host# edit system diameter
```



NOTE: The java-* options have default values that should not be changed unless directed by Juniper Networks Technical Assistance Center (JTAC).

2. If you encounter problems caused by lack of memory, change the maximum memory size available to the Java Runtime Environment (JRE).

```
[edit system diameter]
user@host# set java-heap-size java-heap-size
```

3. Configure the amount of space available to the JRE when the Diameter server starts.

```
[edit system diameter]
user@host# set java-new-size java-new-size
```

4. Configure the garbage collection functionality of the Java Virtual Machine.

```
[edit system diameter]
user@host# set java-garbage-collection-options java-garbage-collection-options
```

5. Specify the protocol for the transport connection.

```
[edit system diameter]
user@host# set protocol [(tcp | sctp)...] 
```

6. (Optional) Specify the local IP addresses that remote peers can use to reach this server.

```
[edit system diameter]
user@host# set local-address [local-address...] 
```

7. (Optional) Specify the port for the server.

```
[edit system diameter]
user@host# set port port 
```

8. (Optional) Specify the fully qualified domain name (FQDN) used to identify this host to its Diameter peers.

```
[edit system diameter]
user@host# set origin-host origin-host 
```

9. (Optional) Specify the realm used to identify this host to its Diameter peers.

```
[edit system diameter]
user@host# set origin-realm origin-realm 
```

The Diameter realm should be configured to the domain name of the origin host. For example, if the FQDN of the host is host.juniper.net, then the realm should be juniper.net.

10. (Optional) Configure the timeout value until which the Diameter server holds unsolicited requests such as Point to Point Protocol (PPP) and Abort Session Request (ASR), and waits for a matching response such as Push Profile Answer (PPA) and Abort Session Answer (ASA). The server discards the responses received after the specified time. The value range is 1–65,565 seconds. The preferred value is 10–30 seconds. By default, the value is set to 25 seconds.

```
[edit system diameter]
user@host# set diameter-server-timeout diameter-server-timeout 
```



NOTE: `diameter-server-timeout` and `reply-timeout` under the `[edit shared sae group configuration driver]` hierarchy should be configured with the same value.

11. (Optional) Specify whether the peer connection is in active mode.

```
[edit system diameter]
user@host# set active-peers
```

**NOTE:**

- Active mode means that the SRC software actively tries to connect to the peer. Make sure the peer you are connecting to supports active peers. The MX Series router does not support active peers. The SRC software can still be configured, but the connection attempts will not work.
- If the peer connection is configured to be in active mode, you must configure the remote peer address for all Diameter peers by using the **address** option under the **[edit shared network diameter peer name]** hierarchy.

12. (Optional) Specify whether the peer connection is in debug mode.

```
[edit system diameter]
user@host# set debug-mode
```

13. (Optional) Configure the load-balancing mode for peer selection when forwarding a request message.

```
[edit system diameter]
user@host# set load-balancing-mode (failover | round-robin)
```

14. (Optional) Configure the log level for the transaction processing log.

```
[edit system diameter]
user@host# set transaction-processing-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

15. (Optional) Configure the log level for the packet tracing log.

```
[edit system diameter]
user@host# set packet-trace-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

16. (Optional) Configure the log level for the peer state machine log.

```
[edit system diameter]
user@host# set peer-state-machine-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

17. (Optional) Configure the log level for the configuration log.

```
[edit system diameter]
user@host# set configuration-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

Configuring the Diameter Client Properties

This procedure configures the client-side adapter of the SRC Diameter server, which handles client connections. Configuration should be necessary only if you encounter performance problems.

Use the following statements to configure the properties for the Diameter client:

```
system diameter client {
  threads threads;
  keep-alive-time keep-alive-time;
}
```

To configure the Diameter client properties:

1. From configuration mode, access the statement for the Diameter client.

```
user@host# edit system diameter client
```

2. (Optional) Specify the number of threads to use.

```
[edit system diameter client]
user@host# set threads threads
```

3. (Optional) Specify the time to wait for new commands.

```
[edit system diameter client]
user@host# set keep-alive-time keep-alive-time
```

Configuring the Diameter Server Properties

Use the following statements to configure the properties for the Diameter server:

```
system diameter server {  
    threads threads;  
    keep-alive-time keep-alive-time;  
}
```

To configure the Diameter server properties:

1. From configuration mode, access the statement for the Diameter server.

```
user@host# edit system diameter server
```

2. (Optional) Specify the minimum number of threads to use.

```
[edit system diameter server]  
user@host# set threads threads
```

3. (Optional) Specify the time to wait for new commands.

```
[edit system diameter server]  
user@host# set keep-alive-time keep-alive-time
```

Configuring Logging Destinations

Use the following configuration statements to configure logging destinations for Diameter:

```
system diameter logger name ...  
  
system diameter logger name file {  
    filter filter;  
    filename filename;  
    rollover-filename rollover-filename;  
    maximum-file-size maximum-file-size;  
}
```

To configure logging destinations to store log messages in a file:

1. From configuration mode, access the statement that configures the name and type of logging destination.

```
user@host# edit system diameter logger name file
```

2. Specify the properties for the logging destination.

```
[edit system diameter logger name file]  
user@host# set ?
```

For more information about configuring properties for the logging destination, see *Configuring Logging Destinations to Store Messages in a File (SRC CLI)*.

- Related Documentation**
- *SRC CLI Commands to Monitor the SRC Diameter Server*
 - To manage services for JSRC peers on MX Series routers, see *Managing Services on MX Series Routers Using the Diameter Application*.

Configuring Diameter Peers (SRC CLI)

Use the following configuration statements to configure the Diameter peers:

```
shared network diameter peer name {
  protocol [(tcp | sctp)...];
  address [address...];
  enforce-source-address;
  local-address local-address;
  connect-timeout connect-timeout;
  watchdog-timeout watchdog-timeout;
  state-machine-timeout state-machine-timeout;
  reconnect-timeout reconnect-timeout;
  port port;
  origin-host origin-host;
  incoming-queue-limit incoming-queue-limit;
  active-peer;
}
```



NOTE: When you commit the Diameter peer configuration, keep in mind the following conditions:

- The origin host, remote peer address, or both should be specified for the Diameter peer.
- If the enforce source address is configured for the Diameter peer, the remote peer address should be specified for the Diameter peer.
- If the peer connection is configured to be in active mode for a particular Diameter peer or globally for all Diameter peers by using the `active-peers` option under the `[edit system diameter]` hierarchy, the remote peer address should be specified for the Diameter peers.

To configure the Diameter peer:

1. From configuration mode, access the statements for the peer.

```
user@host# edit shared network diameter peer name
```

The peer name must be unique.

2. Specify the protocol for the transport connection.

```
[edit shared network diameter peer name]
user@host# set protocol [(tcp | sctp)...]
```

3. (Optional) Specify the addresses of the remote peer. If SCTP is the transport protocol, you can specify multiple addresses. If TCP is the transport protocol, you can specify only a single address.

```
[edit shared network diameter peer name]  
user@host# set address [address...]
```

4. (Optional) Specify whether the remote peer must connect from one of the IP addresses listed by the **address** option.

```
[edit shared network diameter peer name]  
user@host# set enforce-source-address
```

5. (Optional) Specify the local address of the peer.

```
[edit shared network diameter peer name]  
user@host# set local-address local-address
```

6. (Optional) Specify the maximum amount of time allowed for the Diameter peer to respond to a connection request.

```
[edit shared network diameter peer name]  
user@host# set connect-timeout connect-timeout
```

7. (Optional) Specify the watchdog timeout used for the connection to the remote peer.

```
[edit shared network diameter peer name]  
user@host# set watchdog-timeout watchdog-timeout
```

8. (Optional) Specify the Diameter state machine timeout.

```
[edit shared network diameter peer name]  
user@host# set state-machine-timeout state-machine-timeout
```

9. (Optional) Specify the time interval between connection attempts when the peer is in the disconnected state.

```
[edit shared network diameter peer name]  
user@host# set reconnect-timeout reconnect-timeout
```

10. (Optional) Specify the port for the client.

```
[edit shared network diameter peer name]  
user@host# set port port
```

11. (Optional) Specify the identifier for the endpoint that the peer presents during connection establishment.

```
[edit shared network diameter peer name]  
user@host# set origin-host origin-host
```

12. (Optional) Specify the number of messages allowed on the incoming message queue for a peer.

```
[edit shared network diameter peer name]
user@host# set incoming-queue-limit incoming-queue-limit
```

13. (Optional) Specify whether the peer connection is in active mode.

```
[edit shared network diameter peer name]
user@host# set active-peer
```



NOTE: Active mode means that the SRC software actively tries to connect to the peer. Make sure the peer you are connecting to supports active peers. The MX Series router does not support active peers. The SRC software can still be configured, but the connection attempts will not work.

Related Documentation

- [Configuring the Diameter Application \(SRC CLI\) on page 137](#)
- [Viewing SRC Diameter Server State \(SRC CLI\)](#)

Configuring the NAS Groups (SRC CLI)

Tasks to configure the NAS groups are:

- [Configuring NAS Groups on page 145](#)
- [Configuring the NAS Group Device Capabilities \(SRC CLI\) on page 146](#)
- [Classifying Interfaces on page 147](#)
- [Configuring NAS Group Routes on page 148](#)

Configuring NAS Groups

Use the following configuration statements to configure the NAS groups:

```
shared network nas-group name {
  hosted-by [hosted-by...];
  peers [peers...];
  scope [scope...];
  default-peer default-peer;
  update-grace-period update-grace-period;
  initial-ppr-delay initial-ppr-delay;
}
```

To configure the group of peers:

1. From configuration mode, access the configuration statements for the NAS group.


```
user@host# edit shared network nas-group name
```

2. Specify the hosts that instantiate this peer group. If the peer group is an AAA peer group, the SAEs on the listed hosts create device drivers for this peer group.

```
[edit shared network nas-group name]  
user@host# set hosted-by [hosted-by...]
```

3. (Optional) Specify the peers in this NAS group.

```
[edit shared network nas-group name]  
user@host# set peers [peers...]
```

4. (Optional) Specify the service scopes available to subscribers connected to this NAS group.

```
[edit shared network nas-group name]  
user@host# set scope [scope...]
```

5. (Optional) Specify the default peer.

```
[edit shared network nas-group name]  
user@host# set default-peer default-peer
```

6. (Optional) Specify the grace period for interim updates.

```
[edit shared network nas-group name]  
user@host# set update-grace-period update-grace-period
```

7. (Optional) Specify the delay for sending initial Push-Profile-Requests (PPRs) to install policies.

```
[edit shared network nas-group name]  
user@host# set initial-ppr-delay initial-ppr-delay
```

Configuring the NAS Group Device Capabilities (SRC CLI)

The SAE uses user interim accounting requests to keep the user session alive. Some NAS devices do not send user interim accounting requests, which causes the user session to time out in the SAE. To support this type of NAS device, the SAE can use service interim accounting requests to keep the user session alive.

Use the following configuration statements to configure the NAS group device capabilities:

```
shared network nas-group device-capabilities {  
  no-user-interim-update ;  
}
```

To configure the NAS group device capabilities:

1. From configuration mode, access the configuration statements for the NAS group device capabilities.

```
user@host# edit shared network nas-group device-capabilities
```

2. Specify whether to use service interim accounting requests.

```
[edit shared network nas-group device-capabilities]
user@host# set no-user-interim-update
```

- If this option is set, the SAE uses service interim accounting requests to keep the user session alive in the SAE. The SAE can also send user-tracking events to plug-ins driven by SRC interim update interval.
- If this option is not set, the SAE sends user interim tracking events only when it receives a user interim update from the NAS device.

Classifying Interfaces

Use the following configuration statements to define interface classification scripts:

```
shared network nas-group name interface-classifier rule name {
  target target;
}

shared network nas-group name interface-classifier rule name condition name ...

shared network nas-group name interface-classifier rule name script {
  script-value;
  include include;
}
```

A classification script can contain either a target and a condition or a script. If you do not define a script, the classifier must have both a target and a condition.

To define interface classification scripts:

1. From configuration mode, enter the interface classifier configuration for a NAS group.

```
user@host# edit shared network nas-group name interface-classifier
```

2. Create a rule for the classifier. You can create multiple rules for the classifier.

```
[edit shared network nas-group name interface-classifier]
user@host# edit rule name
```

3. Configure either a target or a script for the rule.

- Configure the target for the rule.

```
[edit shared network nas-group name interface-classifier rule name]
```

```
user@host# set target target
```

If you configure a target for the rule, you must configure a match condition. You can create multiple conditions for the rule. See *Interface Classification Conditions*.

```
[edit shared network nas-group name interface-classifier rule name]
user@host# set condition name
```

- Configure the script for the rule.

```
[edit shared network nas-group name interface-classifier rule name]
user@host# edit script
```

(Optional) You can specify a script target.

```
[edit shared network nas-group name interface-classifier rule name script]
user@host# set script-value
```

(Optional) You can include a script that has already been created.

```
[edit shared network nas-group name interface-classifier rule name script]
user@host# set include include
```

Where *include* is a reference to an existing script that is included in the script you are configuring.

Configuring NAS Group Routes

Use the following configuration statements to configure the route for messages:

```
shared network nas-group name routes name term name {
  precedence precedence;
}

shared network nas-group name routes name {
  transaction-variable (request-packet | user-name | realm);
  dictionary-attribute (user-name | user-password | chap-password | nas-ip-address |
    nas-port | service-type | framed-protocol | framed-ip-address | framed-ip-netmask |
    framed-mtu | framed-compression | login-ip-host | callback-number | state |
    vendor-specific | called-station-id | calling-station-id | nas-identifier | login-lat-service
    | login-lat-node | login-lat-group | chap-challenge | nas-port-type | port-limit |
    login-lat-port);
  operator (equals | not_equal | present | not_present | prefix | suffix | range);
  value value;
  low low;
  high high;
}
```

To configure route selection for messages from the SRC Diameter server:

1. From configuration mode, access the configuration statements for route selection.

```
user@host# edit shared network nas-group name routes name
```

2. (Optional) Specify the order by which the route is selected. The route that meets all the matching criteria and has the lowest precedence is selected first. Routes without the precedence defined are considered after those that have the precedence defined. The route with precedence of -1 is the default route. The default route is considered after all the other routes, and only one default route can be defined.

```
[edit shared network nas-group name routes name]
user@host# set precedence precedence
```

3. From configuration mode, access the configuration statements for route selection criteria.

```
user@host# edit shared network nas-group name routes name term name
```

All the criteria must match for this route to be selected.

4. Specify the name of the transaction variable used as the matching criterion.

```
[edit shared network nas-group name routes name term name]
user@host# set transaction-variable (request-packet | user-name | realm)
```

5. (Optional) Specify the name of the dictionary attribute contained in the attribute store. This is applicable only if the transaction variable is request-packet.

```
[edit shared network nas-group name routes name term name]
user@host# set dictionary-attribute (user-name | user-password | chap-password |
nas-ip-address | nas-port | service-type | framed-protocol | framed-ip-address |
framed-ip-netmask | framed-mtu | framed-compression | login-ip-host |
callback-number | state | vendor-specific | called-station-id | calling-station-id |
nas-identifier | login-lat-service | login-lat-node | login-lat-group | chap-challenge
| nas-port-type | port-limit | login-lat-port)
```

6. Specify the operator for criterion matching.

```
[edit shared network nas-group name routes name term name]
user@host# set operator (equals | not_equal | present | not_present | prefix | suffix |
range)
```

7. (Optional) Specify the value to be matched by the target.

```
[edit shared network nas-group name routes name term name]
user@host# set value value
```

8. (Optional) Specify the low end of the range criterion.

```
[edit shared network nas-group name routes name term name]
user@host# set low low
```

9. (Optional) Specify the high end of the range criterion.

```
[edit shared network nas-group name routes name term name]
```

```
user@host# set high high
```

Configuring the SAE to Manage AAA Devices

Use the following configuration statements to configure the AAA device driver:

```
shared sae configuration driver aaa {  
  sae-community-manager sae-community-manager;  
  origin-host origin-host;  
  origin-realm origin-realm;  
  keep-alive-timeout keep-alive-timeout;  
  registry-retry-interval registry-retry-interval;  
  reply-timeout reply-timeout;  
  sequential-message-timeout sequential-message-timeout;  
  transient-session-timeout transient-session-timeout;  
  max-update-interval max-update-interval;  
  update-grace-period update-grace-period;  
  resume-unrecovered;  
  thread-pool-size thread-pool-size;  
  thread-idle-timeout thread-idle-timeout;  
}
```

To configure the AAA device driver:

1. From configuration mode, access the configuration statements for the AAA device driver.

```
user@host# edit shared sae configuration driver aaa
```

2. Specify the name of the community manager.

```
[edit shared sae configuration driver aaa]  
user@host# set sae-community-manager sae-community-manager
```

3. (Optional) Specify the fully qualified domain name used to identify this host.

```
[edit shared sae configuration driver aaa]  
user@host# set origin-host origin-host
```

4. (Optional) Specify the DNS name of the machine used to identify this host.

```
[edit shared sae configuration driver aaa]  
user@host# set origin-realm origin-realm
```

5. (Optional) Specify the keepalive timeout before the registry to a Diameter server expires.

```
[edit shared sae configuration driver aaa]  
user@host# set keep-alive-timeout keep-alive-timeout
```

6. (Optional) Specify the interval between retrying a failed registry to a Diameter server.

```
[edit shared sae configuration driver aaa]
```



```
user@host# set registry-retry-interval registry-retry-interval
```

7. (Optional) Specify the timeout before a request sent to a Diameter server expires.

```
[edit shared sae configuration driver aaa]
user@host# set reply-timeout reply-timeout
```

8. (Optional) Specify the timeout before an expected message expires.

```
[edit shared sae configuration driver aaa]
user@host# set sequential-message-timeout sequential-message-timeout
```

9. (Optional) Specify the timeout before a temporary session expires.

```
[edit shared sae configuration driver aaa]
user@host# set transient-session-timeout transient-session-timeout
```

10. (Optional) Specify the maximum interval between interim updates for a subscriber session.

```
[edit shared sae configuration driver aaa]
user@host# set max-update-interval max-update-interval
```

11. (Optional) Specify the grace period in which to expect an interim update for a subscriber session.

```
[edit shared sae configuration driver aaa]
user@host# set update-grace-period update-grace-period
```

12. (Optional) Specify whether to resume a subscriber session that has failed to recover from a failover.

```
[edit shared sae configuration driver aaa]
user@host# set resume-unrecovered
```

13. (Optional) Specify the number of working threads that process requests.

```
[edit shared sae configuration driver aaa]
user@host# set thread-pool-size thread-pool-size
```

14. (Optional) Specify the timeout for stopping working threads after they become idle.

```
[edit shared sae configuration driver aaa]
user@host# set thread-idle-timeout thread-idle-timeout
```

15. (Optional) Configure the session store parameters for the AAA device driver.

From configuration mode, access the configuration statement that configures the session store for the AAA device driver.

```
user@host# edit shared sae configuration driver aaa session-store
```

For more information about configuring session store parameters, see *Configuring the Session Store Feature (SRC CLI)*.

Configuring AAA Policies (SRC CLI)

Tasks to configure AAA policies are:

- [Configuring AAA Policy Lists on page 152](#)
- [Configuring AAA Policy Rules on page 152](#)
- [Configuring Template Activation Actions on page 152](#)

Configuring AAA Policy Lists

To configure AAA policy lists:

1. From configuration mode, create a policy list. For example, to create a policy list called l1 within a policy group called tiered_aaa:

```
user@host# edit policies group tiered_aaa list l1
```

2. Specify the type of policy list.

```
[edit policies group tiered_aaa list l1]  
user@host# set role aaa
```

3. Specify where the policy is applied on the device.

```
[edit policies group tiered_aaa list l1]  
user@host# set applicability both
```

Configuring AAA Policy Rules

To configure AAA policy rules:

1. From configuration mode, create a policy rule inside a policy list that has already been created and configured. For example, to create a policy rule called r1 within policy list l1:

```
user@host# edit policies group tiered_aaa list l1 rule r1
```

2. Specify the type of policy rule.

```
[edit policies group tiered_aaa list l1 rule r1]  
user@host# set type aaa
```

Configuring Template Activation Actions

Use this action to activate service templates for RADIUS-enabled devices. You can configure template activation actions for AAA policy rules.

The template name and parameters are listed in the SIC service templates.



NOTE: We recommend that the `user_ipMask` and `user_ipAddress` runtime parameters be avoided for activate-on-login services.

Use the following configuration statements to configure a template activation action:

```

policies group name list name rule name template-activation name {
  template-name template-name;
  description description;
}

policies group name list name rule name template-activation name variables name {
  value value;
  type type;
}

```

To configure a template activation action:

1. From configuration mode, enter the template activation action configuration. For example, in this procedure, `ta` is the name of the template activation action.

```
user@host# edit policies group tiered_aaa list l1 rule r1 template-activation ta
```

2. Enter the template name to activate.

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta]
user@host# set template-name template-name
```

3. (Optional) Enter a description for the template activation action.

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta]
user@host# set description description
```

4. From configuration mode, enter the parameters used by the template.

```
user@host# edit policies group tiered_aaa list l1 rule r1 template-activation ta variables
name
```

For example:

```
user@host# edit policies group tiered_aaa list l1 rule r1 template-activation ta variables
upstreamBandwidth
```

5. (Optional) Configure the value for the variable.

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables name]
user@host# set value value
```

For example:

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables
upstreamBandwidth]
user@host# set value rateParameter
```

6. (Optional) Configure the variable type. Variable types are mapped to parameter types.

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables name]  
user@host# set type type
```

For example:

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables  
  upstreamBandwidth]  
user@host# set type rate
```

**Related
Documentation**

- *Configuring Template Activation Actions (SRC CLI)*
- *Before You Configure SRC Policies*
- [Managing Dynamic Services on RADIUS-Enabled Devices on page 35](#)

CHAPTER 6

Example for Subscriber Information Collector

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 155](#)
- [Sample Service Templates on page 165](#)

Example: Basic SIC Group Configuration (SRC CLI)

This sample configuration uses the default SIC group called default-group, and the default SIC server called default-server.

An editing rule called username specifies that if the source, which is the request attribute User-Name, contains the @test.com suffix, the suffix is to be removed, and the resulting value placed in the target, which is the request attribute User-Name. A second editing rule, called vpnid, specifies that the target, which is the SIC variable vpn-id, should be replaced with the value of the source, which is the request attribute NAS-Identifier.

The SIC group (default-group) includes the default device model called default-model, which are both using the default dictionary called radius.

The accounting listener for the SIC listens on port 1813 for incoming accounting events. An upstream network element called netpc is using the default device model called default-model. The netpc network element contains four accounting clients called netpc13, netpc14, netpc15, and netpc16. The IP addresses and shared secrets of these accounting clients are provided as examples only. The outbound transport uses port 0.

The accounting route called test-route specifies that the editing rule called vpnid is to be applied before the request is routed to the accounting target.

[Table 18 on page 155](#) lists the attribute mapping defined between the SIC and the SAE plug-in attributes.

Table 18: Sample Configuration Attribute Associations

SIC Variable or Attribute	SAE Plug-In Attribute
Request-attribute User-Name	Login-name
Request-attribute Calling-Station-Id	Property.calling-station-id

Table 18: Sample Configuration Attribute Associations (*continued*)

SIC Variable or Attribute	SAE Plug-In Attribute
Variable ReceiveTime	Property.session-start-time
Variable UserStatusType	Property.session-state
Request-attribute Framed-IP-Address	User-inet-address

Three log streams are configured, including the default log stream called default-logger, which captures events for the log groups at the event levels listed in [Table 19 on page 156](#).

Table 19: Log Groups and Associated Event Level for Log Stream=default logger

Log Group	Event Level
Administration	Warning
Configuration	Warning
Packet	Debug
PacketTrace	Warning
PacketTraceRaw	Warning
System	Warning

Two additional log streams are configured, called debug-logger and error-logger, which capture events for the log groups at the event levels listed in [Table 20 on page 156](#) and [Table 21 on page 157](#).

Table 20: Log Groups and Associated Event Level for Log Stream=debug-logger

Log Group	Event Level
Administration	Debug
Configuration	Debug
Packet	Debug
PacketTrace	Debug
PacketTraceRaw	Debug
System	Debug

Table 21: Log Groups and Associated Event Level for Log Stream=error-logger

Log Group	Event Level
Administration	Warning
Configuration	Warning
Packet	Warning
PacketTrace	Warning
PacketTraceRaw	Warning
System	Warning

user@host# show slot 0 sic

```
initial {
  directory-connection {
    credentials *****;
    entry-dn l=SIC,ou=staticConfiguration,ou=Configuration,o=Management,o=umc;
    filter (objectClass=*);
    port 389;
    principal cn=umcadmin,o=umc;
    url 127.0.0.1;
  }
}
server {
  name default-server;
}
[edit]
```

user@host# show shared sic group default-group editing

```
username {
  mode replace;
  source {
    request-attribute {
      User-Name {
        remove-suffix @test.com;
      }
    }
  }
  target {
    request-attribute User-Name;
  }
}
vpnid {
  mode replace;
  source {
    request-attribute {
      NAS-Identifier;
    }
  }
}
```

```
    }
    target {
        variable vpn-id;
    }
}
```

[edit]

user@host# **show shared sic group default-group radius**

```
accounting-listener {
    transport {
        1813 {
            connect-timeout 1000;
            connections-per-thread 15;
            disconnect-timeout 1000;
            port 1813;
        }
    }
}
network-element netpc {
    upstream {
        model default-model;
        accounting-client {
            netpc13 {
                accounting-secret secret;
                address 10.227.6.213;
            }
            netpc14 {
                accounting-secret secret;
                address 10.227.6.214;
            }
            netpc15 {
                accounting-secret secret;
                address 10.227.6.215;
            }
            netpc16 {
                accounting-secret secret;
                address 10.227.6.216;
            }
        }
    }
}
outbound-transport {
    default-outbound-transport {
        connect-timeout 1000;
        connections-per-thread 15;
        disconnect-timeout 1000;
        port 0;
    }
}
```

[edit]

user@host# **show shared sic group default-group dictionary radius**

```
attribute ARAP-Challenge-Response {
    radius {
        format octets;
        type 84;
    }
}
```



```
}
attribute ARAP-Features {
  radius {
    format octets;
    type 71;
  }
}
attribute ARAP-Password {
  radius {
    format octets;
    type 70;
  }
}
attribute Proxy-State {
  radius {
    format string;
    type 33;
  }
}
attribute Reply-Message {
  radius {
    format string;
    type 18;
  }
}
attribute Service-Type {
  radius {
    constant Administrative {
      6;
    }
    constant Authenticate-Only {
      8;
    }
    constant Authorize-Only {
      17;
    }
    constant Call-Check {
      10;
    }
    constant Callback-Administrative {
      11;
    }
    constant Callback-Framed {
      4;
    }
    constant Callback-Login {
      3;
    }
    constant Callback-NAS-Prompt {
      9;
    }
    constant Fax {
      13;
    }
    constant Framed {
      2;
    }
    constant IAPP-AP-Check {
      16;
    }
  }
}
```

```
        constant IAPP-Register {
            15;
        }
        constant Login {
            1;
        }
        constant Modem-Relay {
            14;
        }
        constant NAS-Prompt {
            7;
        }
        constant Outbound {
            5;
        }
        constant Voice {
            12;
        }
        format integer;
        type 6;
    }
}
attribute Session-Timeout {
    radius {
        format integer;
        type 27;
    }
}
attribute State {
    radius {
        format string;
        type 24;
    }
}
attribute TeliaSonera-Chargeable-User-Id {
    radius {
        format string;
        type 192;
        vendor-id 15297;
    }
}
attribute TeliaSonera-Location-Info {
    radius {
        format string;
        type 194;
        vendor-id 15297;
    }
}
attribute TeliaSonera-Location-Name {
    radius {
        format string;
        type 195;
        vendor-id 15297;
    }
}
attribute TeliaSonera-Operator-Name {
    radius {
        format string;
        type 193;
        vendor-id 15297;
    }
}
```

```

}
attribute TeliaSonera-Visited-Operator-ID {
    radius {
        format string;
        type 196;
        vendor-id 15297;
    }
}
attribute Termination-Action {
    radius {
        constant Default {
            0;
        }
        constant RADIUS-Request {
            1;
        }
        format integer;
        type 29;
    }
}
attribute Tunnel-Assignment-ID {
    radius {
        format string;
        tagged;
        type 82;
    }
}
attribute Tunnel-Client-Auth-ID {
    radius {
        format string;
        tagged;
        type 90;
    }
}
attribute Tunnel-Client-Endpoint {
    radius {
        format string;
        tagged;
        type 66;
    }
}
attribute Tunnel-Medium-Type {
    radius {
        constant 802 {
            6;
        }
        constant ATM {
            3;
        }
        constant Appletalk {
            12;
        }
        constant BBN-1822 {
            5;
        }
        constant Banyan-Vines {
            14;
        }
        constant Decnet-IV {
            13;
        }
    }
}

```

```
        constant E.163 {
            7;
        }
        constant E.164 {
            8;
        }
        constant E.164-NSAP-subaddress {
            15;
        }
        constant F.69 {
            9;
        }
        constant Frame-Relay {
            4;
        }
        constant IP {
            1;
        }
        constant IPX {
            11;
        }
        constant X.121 {
            10;
        }
        constant X.25 {
            2;
        }
        format integer;
        tagged;
        type 65;
    }
}
attribute Tunnel-Password {
    radius {
        format string;
        salt-encrypt;
        tagged;
        type 69;
    }
}
attribute Tunnel-Preference {
    radius {
        format integer;
        tagged;
        type 83;
    }
}
attribute Tunnel-Private-Group-ID {
    radius {
        format string;
        tagged;
        type 81;
    }
}
attribute Tunnel-Server-Auth-ID {
    radius {
        format string;
        tagged;
        type 91;
    }
}
```

```
attribute Tunnel-Server-Endpoint {
  radius {
    format string;
    tagged;
    type 67;
  }
}
attribute Tunnel-Type {
  radius {
    constant AH {
      6;
    }
    constant ATMP {
      4;
    }
    constant DVS {
      11;
    }
    constant ESP {
      9;
    }
    constant GRE {
      10;
    }
    constant IP-IP {
      7;
    }
    constant IP-IP-Tunneling {
      12;
    }
    constant L2F {
      2;
    }
    constant L2TP {
      3;
    }
    constant MIN-IP-IP {
      8;
    }
    constant PPTP {
      1;
    }
    constant VLAN {
      13;
    }
    constant VTP {
      5;
    }
    format integer;
    tagged;
    type 64;
  }
}
attribute User-Name {
  radius {
    format string;
    type 1;
  }
}
attribute User-Password {
  radius {
```

```
        format string;
        type 2;
    }
}
user@host# show default-model
dictionary radius;

*****

user@host# show shared sic group default-group server
default-server {
    accounting-route {
        test-route {
            editing {
                vpnid;
            }
            target {
                accounting-method default-method;
            }
        }
        default-route {
            target {
                accounting-method default-method;
            }
        }
    }
}
logger {
    debug-logger {
        file {
            filename sic_debug;
            filter /debug-error;
            flush-after-writes;
            maximum-file-size 0;
            prepend-message-header;
            rollover-interval 86400;
        }
        group {
            administration events debug;
            configuration events debug;
            packet events debug;
            packet-trace events debug;
            packet-trace-raw events debug;
            system events debug;
        }
    }
}
default-logger {
    file {
        filename sic;
        filter customized;
        flush-after-writes;
        maximum-file-size 0;
        prepend-message-header;
        rollover-interval 86400;
    }
    group {
        administration events warning;
        configuration events warning;
        packet events debug;
        packet-trace events warning;
        packet-trace-raw events warning;
        system events warning;
    }
}
```

```

    }
  }
  error-logger {
    file {
      filename sic_error;
      filter /error;
      flush-after-writes;
      maximum-file-size 0;
      prepend-message-header;
      rollover-interval 86400;
    }
    group {
      administration events warning;
      configuration events warning;
      packet events warning;
      packet-trace events warning;
      packet-trace-raw events warning;
      system events warning;
    }
  }
}
}
[edit]

```

Related Documentation

- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 12](#)
- [Accounting Method and Target \(SRC CLI\) on page 13](#)
- [SIC Editing Rules \(SRC CLI\) on page 17](#)
- [RADIUS and Diameter Configuration for the SIC Overview \(SRC CLI\) on page 21](#)

Sample Service Templates

We provide templates for Juniper Networks E Series Broadband Services Routers running JunosE Software release 7.2 or later and for Cisco routers running Cisco IOS Release 12.2SB. These templates include sample global and service templates that you can modify for your specific environment. Each sample includes a global service template and at least one service template.

Juniper Networks Routers Service Template

This example shows the complete Juniper Networks service template configuration. This template supports Juniper Networks E Series Broadband Services Routers running JunosE Software release 7.2 or later.

```

user@host# show device-template juniper-router-junose-7.2-plus
capabilities {
  capability activation {
    Both;
  }
  capability bundle {
    Single;
  }
  capability modification {
    false;
  }
}

```

```

    }
  }
  model juniper-router-junos-7.2-plus;
  global-template {
    description 'global section';
    mode abort-session {
      attributes {
        attribute Acct-Session-Id {
          required {
            name Acct-Session-Id;
          }
        }
      }
    }
    variable RadiusCode {
      type integer;
      value 40;
    }
  }
  mode accounting {
    attributes;
  }
  mode authentication {
    attributes;
  }
}
service-template content_provider_tiered {
  description 'content_provider_tiered service';
  mode activation {
    attributes {
      attribute Acct-Session-Id {
        required {
          name Acct-Session-Id;
        }
      }
    }
    tagged-group {
      attribute Unisphere-Activate-Service {
        parameterized {
          format 'content_provider_tiered\\($(contentProviderAddress),$(contentProviderMask),$(subscriberAddress),$(subscriberMask),$(upstreamBandwidth),$(downstreamBandwidth)\\)';
          name Unisphere-Activate-Service;
        }
      }
      attribute Unisphere-Service-Stats {
        default {
          name Unisphere-Service-Stats;
          value 2;
        }
      }
    }
  }
  variable RadiusCode {
    type integer;
    value 43;
  }
}
mode deactivation {
  attributes {
    attribute Acct-Session-Id {
      required {
        name Acct-Session-Id;
      }
    }
  }
}

```



```

    }
  }
  attribute Unisphere-Deactivate-Service {
    parameterized {
      format 'content_provider_tiered\\($(contentProviderAddress),$(contentP
roviderMask),$(subscriberAddress),$(subscriberMask),$(upstreamBandwidth),$(downs
treamBandwidth)\\)';
      name Unisphere-Deactivate-Service;
    }
  }
}
variable RadiusCode {
  type integer;
  value 43;
}
}
mode initial-authorization {
  attributes {
    tagged-group {
      attribute Unisphere-Activate-Service {
        parameterized {
          format 'content_provider_tiered\\($(contentProviderAddress),$(conten
tProviderMask),$(subscriberAddress),$(subscriberMask),$(upstreamBandwidth),$(dow
nstreamBandwidth)\\)';
          name Unisphere-Activate-Service;
        }
      }
      attribute Unisphere-Service-Stats {
        default {
          name Unisphere-Service-Stats;
          value 2;
        }
      }
    }
  }
}
mode service-correlation-id {
  attributes {
    attribute Juniper-Service-Correlation-Id {
      parameterized {
        format 'content_provider_tiered\\($(contentProviderAddress),$(contentP
roviderMask),$(subscriberAddress),$(subscriberMask),$(upstreamBandwidth),$(downs
treamBandwidth)\\)';
        name Juniper-Service-Correlation-Id;
      }
    }
  }
}
}
service-template internet_tiered {
  description 'internet_tiered service';
  mode activation {
    attributes {
      attribute Acct-Session-Id {
        required {
          name Acct-Session-Id;
        }
      }
    }
    tagged-group {
      attribute Unisphere-Activate-Service {
        parameterized {

```

```

        format 'internet_tiered\\$(upstreamBandwidth),$(downstreamBandwidth
)\\)';
        name Unisphere-Activate-Service;
    }
}
attribute Unisphere-Service-Stats {
    default {
        name Unisphere-Service-Stats;
        value 2;
    }
}
}
variable RadiusCode {
    type integer;
    value 43;
}
}
mode deactivation {
    attributes {
        attribute Acct-Session-Id {
            required {
                name Acct-Session-Id;
            }
        }
        attribute Unisphere-Deactivate-Service {
            parameterized {
                format 'internet_tiered\\$(upstreamBandwidth),$(downstreamBandwidth)\\
)\\)';
                name Unisphere-Deactivate-Service;
            }
        }
    }
    variable RadiusCode {
        type integer;
        value 43;
    }
}
mode initial-authorization {
    attributes {
        tagged-group {
            attribute Unisphere-Activate-Service {
                parameterized {
                    format 'internet_tiered\\$(upstreamBandwidth),$(downstreamBandwidth
)\\)';
                    name Unisphere-Activate-Service;
                }
            }
            attribute Unisphere-Service-Stats {
                default {
                    name Unisphere-Service-Stats;
                    value 2;
                }
            }
        }
    }
}
mode service-correlation-id {
    attributes {
        attribute Juniper-Service-Correlation-Id {
            parameterized {

```

```

        format 'internet_tiered\\($(upstreamBandwidth),$(downstreamBandwidth))\
    \)';
        name Juniper-Service-Correlation-Id;
    }
}
}
}
}
service-template guided_entrance {
    description 'guided_entrance service';
    mode activation {
        attributes {
            attribute Acct-Session-Id {
                required {
                    name Acct-Session-Id;
                }
            }
            tagged-group {
                attribute Unisphere-Activate-Service {
                    parameterized {
                        format 'guided_entrance\\($(redirectAddress),$(redirectPort),$(redirectRemainingUrl),$(originalAddress),$(originalMask),$(originalPort))\)';
                        name Unisphere-Activate-Service;
                    }
                }
                attribute Unisphere-Service-Stats {
                    default {
                        name Unisphere-Service-Stats;
                        value 1;
                    }
                }
            }
        }
        variable RadiusCode {
            type integer;
            value 43;
        }
    }
    mode deactivation {
        attributes {
            attribute Acct-Session-Id {
                required {
                    name Acct-Session-Id;
                }
            }
            attribute Unisphere-Deactivate-Service {
                parameterized {
                    format 'guided_entrance\\($(redirectAddress),$(redirectPort),$(redirectRemainingUrl),$(originalAddress),$(originalMask),$(originalPort))\)';
                    name Unisphere-Deactivate-Service;
                }
            }
        }
        variable RadiusCode {
            type integer;
            value 43;
        }
    }
    mode initial-authorization {
        attributes {
            tagged-group {

```

```

        attribute Unisphere-Activate-Service {
            parameterized {
                format 'guided_entrance\\($redirectAddress),$($redirectPort),$($redirectRemainingUrl),$($originalAddress),$($originalMask),$($originalPort)\\';
                name Unisphere-Activate-Service;
            }
        }
        attribute Unisphere-Service-Stats {
            default {
                name Unisphere-Service-Stats;
                value 1;
            }
        }
    }
}
mode service-correlation-id {
    attributes {
        attribute Juniper-Service-Correlation-Id {
            parameterized {
                format 'guided_entrance\\($redirectAddress),$($redirectPort),$($redirectRemainingUrl),$($originalAddress),$($originalMask),$($originalPort)\\';
                name Juniper-Service-Correlation-Id;
            }
        }
    }
}
}
vendor juniper;

[edit shared sic group test-group]
user@host#

```

Cisco Router Service Template

This example shows the complete Cisco router service template configuration. This template supports Cisco routers running Cisco IOS Release 12.2SB.

```

user@host# show device-template cisco-router-ios-12.2-sb
capabilities {
    capability activation {
        Both;
    }
    capability bundle {
        None;
    }
    capability modification {
        false;
    }
}
model cisco-router-ios-12.2-sb;
global-template {
    description 'global section';
    mode abort-session {
        attributes {
            attribute Acct-Session-Id {
                required {
                    name Acct-Session-Id;
                }
            }
        }
    }
}

```

```

    }
    variable RadiusCode {
        type integer;
        value 40;
    }
}
mode accounting {
    attributes;
}
mode authentication {
    attributes;
}
}
service-template content_provider_tiered {
    description 'content_provider_tiered service';
    mode activation {
        attributes {
            attribute Acct-Session-Id {
                required {
                    name Acct-Session-Id;
                }
            }
            attribute Cisco-SSG-Command-Code {
                override {
                    name Cisco-SSG-Command-Code;
                    value '{hex}0B 63 6F 6E 74 65 6E 74 5F 70 72 6F 76 69 64 65 72 5F 74
69 65 72 65 64';
                }
            }
            attribute Cisco-AVPair-1 {
                parameterized {
                    format 'ip:inac1#10=permit ip any $(subscriberAddress)
$(subscriberMask)';
                    name Cisco-AVPair;
                }
            }
            attribute Cisco-AVPair-2 {
                parameterized {
                    format 'ip:outac1#20=permit ip $(contentProviderAddress)
$(contentProviderMask) any';
                    name Cisco-AVPair;
                }
            }
            attribute Cisco-AVPair-3 {
                override {
                    name Cisco-AVPair;
                    value 'ip:traffic-class=in access-group name 10 priority 10';
                }
            }
            attribute Cisco-AVPair-4 {
                override {
                    name Cisco-AVPair;
                    value 'ip:traffic-class=in default drop';
                }
            }
            attribute Cisco-AVPair-5 {
                override {
                    name Cisco-AVPair;
                    value 'ip:traffic-class=out access-group name 20 priority 10';
                }
            }
        }
    }
}

```

```

        attribute Cisco-AVPair-6 {
            override {
                name Cisco-AVPair;
                value 'ip:traffic-class=out default drop';
            }
        }
        attribute Cisco-SSG-Service-Info {
            parameterized {
                format 'QU;$(upstreamBandwidth);;;D;$(downstreamBandwidth);';
                name Cisco-SSG-Service-Info;
            }
        }
        attribute Cisco-AVPair-7 {
            override {
                name Cisco-AVPair;
                value accounting-list=default;
            }
        }
    }
    variable RadiusCode {
        type integer;
        value 43;
    }
}
mode deactivation {
    attributes {
        attribute Acct-Session-Id {
            required {
                name Acct-Session-Id;
            }
        }
        attribute Cisco-SSG-Command-Code {
            override {
                name Cisco-SSG-Command-Code;
                value '{hex}0C 63 6F 6E 74 65 6E 74 5F 70 72 6F 76 69 64 65 72 5F 74
69 65 72 65 64';
            }
        }
    }
    variable RadiusCode {
        type integer;
        value 43;
    }
}
mode service-correlation-id {
    attributes {
        attribute Juniper-Service-Correlation-Id {
            override {
                name Juniper-Service-Correlation-Id;
                value Ncontent_provider_tiered;
            }
        }
    }
}
mode service-profile-download {
    attributes {
        attribute User-Name {
            override {
                name User-Name;
                value content_provider_tiered;
            }
        }
    }
}

```

```

    }
  }
}
service-template internet_tiered {
  description 'internet_tiered service';
  mode activation {
    attributes {
      attribute Acct-Session-Id {
        required {
          name Acct-Session-Id;
        }
      }
      attribute Cisco-SSG-Command-Code {
        override {
          name Cisco-SSG-Command-Code;
          value '{hex}0B 69 6E 74 65 72 6E 65 74 5F 74 69 65 72 65 64';
        }
      }
      attribute Cisco-SSG-Service-Info {
        parameterized {
          format 'QU;$(upstreamBandwidth);;D;$(downstreamBandwidth);;';
          name Cisco-SSG-Service-Info;
        }
      }
      attribute Cisco-AVPair {
        override {
          name Cisco-AVPair;
          value accounting-list=default;
        }
      }
    }
  }
  variable RadiusCode {
    type integer;
    value 43;
  }
}
mode deactivation {
  attributes {
    attribute Acct-Session-Id {
      required {
        name Acct-Session-Id;
      }
    }
    attribute Cisco-SSG-Command-Code {
      override {
        name Cisco-SSG-Command-Code;
        value '{hex}0C 69 6E 74 65 72 6E 65 74 5F 74 69 65 72 65 64';
      }
    }
  }
  variable RadiusCode {
    type integer;
    value 43;
  }
}
mode service-correlation-id {
  attributes {
    attribute Juniper-Service-Correlation-Id {
      override {
        name Juniper-Service-Correlation-Id;
      }
    }
  }
}

```

```
        value Ninternet_tiered;
    }
}
}
mode service-profile-download {
    attributes {
        attribute User-Name {
            override {
                name User-Name;
                value internet_tiered;
            }
        }
    }
}
}
service-template guided_entrance {
    description 'guided_entrance service';
    mode activation {
        attributes {
            attribute Acct-Session-Id {
                required {
                    name Acct-Session-Id;
                }
            }
            attribute Cisco-SSG-Command-Code {
                override {
                    name Cisco-SSG-Command-Code;
                    value '{hex}0B 67 75 69 64 65 64 5F 65 6E 74 72 61 6E 63 65';
                }
            }
            attribute Cisco-AVPair-1 {
                parameterized {
                    format 'ip:inac1#10=permit ip any $(originalAddress) $(originalMask)';

                    name Cisco-AVPair;
                }
            }
            attribute Cisco-AVPair-2 {
                parameterized {
                    format 'ip:inac1#20=permit tcp any eq $(originalPort)';
                    name Cisco-AVPair;
                }
            }
            attribute Cisco-AVPair-3 {
                override {
                    name Cisco-AVPair;
                    value 'ip:traffic-class=in access-group name 10 priority 10';
                }
            }
            attribute Cisco-AVPair-4 {
                override {
                    name Cisco-AVPair;
                    value 'ip:traffic-class=in access-group name 20 priority 10';
                }
            }
            attribute Cisco-AVPair-5 {
                parameterized {
                    format 'ip:l4redirect=redirect to ip $(redirectAddress) port
$(redirectPort)';
                    name Cisco-AVPair;
                }
            }
        }
    }
}
```



```

    }
  }
}
variable RadiusCode {
  type integer;
  value 43;
}
}
mode deactivation {
  attributes {
    attribute Acct-Session-Id {
      required {
        name Acct-Session-Id;
      }
    }
    attribute Cisco-SSG-Command-Code {
      override {
        name Cisco-SSG-Command-Code;
        value '{hex}0C 67 75 69 64 65 64 5F 65 6E 74 72 61 6E 63 65';
      }
    }
  }
}
variable RadiusCode {
  type integer;
  value 43;
}
}
mode service-correlation-id {
  attributes {
    attribute Juniper-Service-Correlation-Id {
      override {
        name Juniper-Service-Correlation-Id;
        value Nguided_entrance;
      }
    }
  }
}
mode service-profile-download {
  attributes {
    attribute User-Name {
      override {
        name User-Name;
        value guided_entrance;
      }
    }
  }
}
}
vendor cisco;

[edit shared sic group test-group]
user@host#

```

Related Documentation

- [Device and Service Template Configuration Overview \(SRC CLI\) on page 95](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 103](#)
- [Configuring SIC Service Templates \(SRC CLI\) on page 105](#)

- [Configuring Global Service Templates \(SRC CLI\) on page 122](#)

PART 3

Administration

- [Routine Monitoring on page 179](#)

CHAPTER 7

Routine Monitoring

- [Viewing Statistics for RADIUS Client Accounting Requests \(SRC CLI\) on page 179](#)
- [Viewing Statistics for RADIUS Client Authentication Requests \(SRC CLI\) on page 179](#)
- [Viewing RADIUS Host Statistics for Accounting Transactions \(SRC CLI\) on page 180](#)
- [Viewing RADIUS Host Statistics for Authentication Transactions \(SRC CLI\) on page 180](#)
- [Viewing RADIUS Target Statistics for Accounting Requests \(SRC CLI\) on page 181](#)
- [Viewing RADIUS Target Statistics for Authentication Requests \(SRC CLI\) on page 181](#)
- [Viewing Diameter Host Statistics \(SRC CLI\) on page 182](#)
- [Viewing Diameter Peer Statistics \(SRC CLI\) on page 182](#)

Viewing Statistics for RADIUS Client Accounting Requests (SRC CLI)

Purpose View RADIUS client statistics for accounting requests. Statistics are presented for any client from which the server has received packets.

Action `user@host> show sic statistics radius client accounting`

Related Documentation

- [Viewing Statistics for RADIUS Client Authentication Requests \(SRC CLI\) on page 179](#)
- [Viewing RADIUS Host Statistics for Accounting Transactions \(SRC CLI\) on page 180](#)
- [Viewing RADIUS Target Statistics for Accounting Requests \(SRC CLI\) on page 181](#)

Viewing Statistics for RADIUS Client Authentication Requests (SRC CLI)

Purpose View RADIUS client statistics for authentication requests. Statistics are presented for any client from which the server has received packets.

Action user@host> **show sic statistics radius client authentication**

- Related Documentation**
- [Viewing RADIUS Host Statistics for Authentication Transactions \(SRC CLI\) on page 180](#)
 - [Viewing RADIUS Target Statistics for Authentication Requests \(SRC CLI\) on page 181](#)
 - [Viewing Statistics for RADIUS Client Accounting Requests \(SRC CLI\) on page 179](#)

Viewing RADIUS Host Statistics for Accounting Transactions (SRC CLI)

Purpose View RADIUS host statistics for accounting transactions, as well as server runtime and packet error statistics.

Action user@host> **show sic statistics radius host accounting**

```
RADIUS Host Accounting Statistics
Name as accounting server      SIC
Up time:                      6791110
Reset time:                    0
Server status:                 4
Requests:                      1660
Invalid requests:              0
Duplicate requests:            0
Responses:                     1660
Malformed requests:           0
Bad authenticators:            0
Packets dropped:                0
No records:                    0
Packets of unknown types:      0
Response from invalid addresses: 0
Name as accounting client      SIC
```

- Related Documentation**
- [Viewing Statistics for RADIUS Client Accounting Requests \(SRC CLI\) on page 179](#)
 - [Viewing RADIUS Target Statistics for Accounting Requests \(SRC CLI\) on page 181](#)
 - [Viewing RADIUS Host Statistics for Authentication Transactions \(SRC CLI\) on page 180](#)

Viewing RADIUS Host Statistics for Authentication Transactions (SRC CLI)

Purpose View RADIUS host statistics for authentication transactions, as well as server runtime and packet error statistics.

Action user@host> **show sic statistics radius host authentication**

```

RADIUS Host Authentication Statistics
Name as authentication server      SIC
Up time:                          6791110
Reset time:                       0
Server status:                    4
Requests:                         1660
Invalid requests:                 0
Duplicate requests:               0
Access accepts:                   1620
Access rejects:                   20
Access challenges:                20
Malformed requests:              0
Bad authenticators:               0
Packets dropped:                  0
No records:                       0
Packets of unknown types:         0
Response from invalid addresses: 0
Name as authentication client     SIC

```

- Related Documentation**
- [Viewing RADIUS Host Statistics for Accounting Transactions \(SRC CLI\) on page 180](#)
 - [Viewing Statistics for RADIUS Client Authentication Requests \(SRC CLI\) on page 179](#)
 - [Viewing RADIUS Target Statistics for Authentication Requests \(SRC CLI\) on page 181](#)

Viewing RADIUS Target Statistics for Accounting Requests (SRC CLI)

Purpose View RADIUS target statistics for accounting requests.

Action user@host> **show sic statistics radius target accounting**

- Related Documentation**
- [Viewing RADIUS Target Statistics for Authentication Requests \(SRC CLI\) on page 181](#)
 - [Viewing Statistics for RADIUS Client Accounting Requests \(SRC CLI\) on page 179](#)
 - [Viewing RADIUS Target Statistics for Authentication Requests \(SRC CLI\) on page 181](#)

Viewing RADIUS Target Statistics for Authentication Requests (SRC CLI)

Purpose View RADIUS target statistics for authentication requests. Statistics are available for RADIUS dynamic authorization and authentication targets that are defined in the server.

Action user@host> **show sic statistics radius target authentication**
user@host> show sic statistics radius target authentication host 10.1.2.3
RADIUS Target Authentication Statistics
Index 0
Address Type Unknown
Address /10.1.2.3
Port 0
Round trip time: 0
Requests: 0
Retransmitted requests: 0
Access accepts: 0
Access rejects: 0
Access challenges: 0
Malformed responses: 0
Bad authenticators: 0
Pending requests: 0
Timeouts: 0
Packets of unknown types: 0
Packets dropped: 0
Counter Discontinuity: 0

user@host>

- Related Documentation**
- [Viewing RADIUS Target Statistics for Accounting Requests \(SRC CLI\) on page 181](#)
 - [Viewing RADIUS Host Statistics for Authentication Transactions \(SRC CLI\) on page 180](#)
 - [Viewing Statistics for RADIUS Client Authentication Requests \(SRC CLI\) on page 179](#)

Viewing Diameter Host Statistics (SRC CLI)

Purpose View Diameter host statistics, including server runtime statistics and global summary statistics.

Action user@host> **show sic statistics diameter host**

- Related Documentation**
- [Viewing Diameter Peer Statistics \(SRC CLI\) on page 182](#)
 - [Viewing RADIUS Host Statistics for Accounting Transactions \(SRC CLI\) on page 180](#)

Viewing Diameter Peer Statistics (SRC CLI)

Purpose Display Diameter peer statistics. These statistics include:

- Connection-related statistics—Statistics related to the connection between the server and the peer.
- Request/Answer statistics—Statistics related to Diameter Request and Diameter Answer messages between the server and the peer.

- Packet error statistics—Statistics related to Diameter errors and message receipt failures.

Action `user@host> show sic statistics diameter peer name name`

Specify the name of the Diameter peer to display statistics; if omitted, statistics related to all Diameter peers are displayed.

Related Documentation • [Viewing Diameter Host Statistics \(SRC CLI\) on page 182](#)

