JUNIPER NETWORKS

Service Management on Third Party Devices

Published: 2013-12-04

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Service Management on Third Party Devices

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.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

List of Figures

List of Tables

About the Documentation

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:

- C Series

Documentation Conventions

Table 1 on page xii 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.

Documentation Conventions

Table 1 on page xii defines the notice icons used in this guide. Table 3 on page xii 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.

Table 3: Text Conventions

Convention	Description	Examples
Bold text like this	• Represents keywords, scripts, and tools in text. • Represents a GUI element that the user selects, clicks, checks, or clears.	• Specify the keyword **exp-msg**. • Run the **install.sh** script. • Use the **pkgadd** tool. • To cancel the configuration, click **Cancel**.

Table 3: Text Conventions *(continued)*

Bold text like this	Represents text that the user must type.	**user@host# set cache-entry-age** *cache-entry-age*

`Fixed-width text like this`	Represents information as displayed on your terminal's screen, such as CLI commands in output displays.	`nic-locators {`
		`  login {`
		`  resolution {`
		`  resolver-name /realms/`
		`  login/A1;`
		`  key-type LoginName;`
		`  value-type SaeId;`
		`  }`

Regular sans serif typeface	• Represents configuration statements. • Indicates SRC CLI commands and options in text. • Represents examples in procedures. • Represents URLs.	• **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
Italic sans serif typeface	Represents variables in SRC CLI commands.	**user@host# set local-address** ***local-address***
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.
Italic typeface	• Emphasizes words. • Identifies book names. • Identifies distinguished names. • Identifies files, directories, and paths in text but not in command examples.	• There are two levels of access: *user* and *privileged*. • *SRC-PE Getting Started Guide.* • *o=Users, o=UMC* • The */etc/default.properties* file.
Backslash	At the end of a line, indicates that the text wraps to the next line.	Plugin.radiusAcct-1.class=\ net.juniper.smgt.sae.plugin\ RadiusTrackingPluginEvent
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.)	diagnostic | line

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to **techpubs-comments@juniper.net**, or fill out the documentation feedback form at

https://www.juniper.net/cgi-bin/docbugreport/ . If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release 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 JNASC 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: https://www.juniper.net/alerts/
- 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

CHAPTER 1

Software Features Overview

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)	• 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)	Used in conjunction with the MX Series router running the packet-triggered subscribers and policy control (PTSP) solution, the SIC listens for RADIUS accounting events from IP edge devices (accounting clients) and stores them in the Session State Registrar (SSR), or 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.

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

Component	Description
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.
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 Gx interface to mediate between the PCRF and Juniper Networks routers like the E Series Broadband Services routers and MX Series routers. The Gx interface on the SRC 3GPP gateway communicates with the PCRF 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	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. 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.
Repository	
Directory	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. For large subscriber databases, you must supply your own directory.
Session State Registrar (SSR)	The SSR is a stateless, highly reliable and highly available database cluster. When used in conjunction with an MX Series router running the packet-triggered subscribers and policy control (PTSP) solution, the SSR stores the IP edge attachment subscriber sessions data learned from IP edge devices in the centralized SSR database.
SRC Configuration and Management Tools	

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

Component	Description
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.
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.

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

Component	Description
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	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. 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.
Demonstration Applications (available on the Juniper Networks Web site)	
Enterprise Audit Plug-In	Defines a callback interface, which receives events when IT managers complete specified operations.
Enterprise Manager Portal	Allows service providers to provision services for enterprise subscribers on routers running JunosE or Junos OS and allows IT managers to manage services. 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 all IT managers to make requests about public IP addresses through the Enterprise Manager Portal.

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

Component	Description
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

CoA Script Service

COA Script Service Overview

The service activation engine (SAE) can use change-of-authorization (COA) messages to manage services for a specific subscriber session. The COA script service allows the SAE to exchange COA messages with third-party devices that do not support Common Open Policy Service (COPS) protocol to activate or deactivate services for specific subscriber sessions. When the SAE activates a COA script service session, the session sends COA messages to a RADIUS-enabled device. This method uses RADIUS attributes and RADIUS vendor-specific attributes (VSAs) to identify a subscriber session whose services are to be activated or deactivated.

Related Documentation

- Configuring COA Script Services on page 21
- Configuring Subscriptions to the COA Script Service on page 24
- Configuring Monitoring Agent to Receive RADIUS Accounting Messages on page 22
- Parameters for Sample COA Script Service on page 9
- Example: Using the Sample COA Script Service on page 27

Parameters for Sample COA Script Service

Table 5 on page 9 lists the parameters specified by the sample COA script service, which is the */SDK/scriptServices/coa/ldif/BOD1M.ldif* file in the **SDK+AppSupport+Demos+Samples.tar.gz** file. You can use the sample script service as a starting point.

Table 5: Parameter Substitutions for COA Services

Parameter Name	Description
dynClientIp	IP address of the third-party device.
dynClientPort	UDP port number of the third-party device.

Table 5: Parameter Substitutions for COA Services *(continued)*

Parameter Name	Description
dynServerIp	IP address of the C Series Controller.
dynServerPort	UDP port number of the C Series Controller.
dynSecret	Shared secret between RADIUS server and RADIUS client.
dynRetry	Number of retries for sending COA messages when no RADIUS response is received. The retry interval is 3 seconds.

dynConfig	Content of service definition in the format <action>. <radiusAttributeName>=<pluginEventAttribute>\n
	• action—Action that is executed on packet content (attribute):
	  ◦ start
	  ◦ stop
	  ◦ start-stop
	• radiusAttributeName—Valid RADIUS attribute specified as follows:
	  ◦ Standard RADIUS attribute name or number
	  ◦ Third-party VSA in the format vendor-specific.<vendor#>.<vsa#>.string
	• pluginEventAttribute—Valid expression in the format:
	  ◦ Python expression
	  ◦ <commandCode><serviceName>; the entire expression must be enclosed in single quotation marks and you must use three backslashes (\\\) to escape the backslash that starts a <commandCode>
	For example: \x0b would be replaced by \\\\x0b
	• \n—New-line character included between the lines of a configuration containing multiple lines; the entire configuration must be enclosed in quotation marks.
	For example:
	start-stop.Acct-Session-Id = ifSessionId
	" start-stop.Acct-Session-Id=ifSessionId\nstart.vendor-specific.9.252.string='\\\\x0bBOD1M'\nstop.vendor-specific.9.252.string='\\\\x0cBOD1M'\n"

You can also configure dynamic RADIUS requests with the sendDynamicRadius method of the ServiceSessionInfo interface (see "Defining RADIUS Attributes for COA Requests with the API" on page 24).

Related Documentation

- COA Script Service Overview on page 9
- Configuring Monitoring Agent to Receive RADIUS Accounting Messages on page 22
- Creating the COA Script Service (SRC CLI) on page 22
- Configuring COA Script Services on page 21
- Example: Using the Sample COA Script Service on page 27

CHAPTER 3

Dynamic Authorization

Managing Dynamic Services

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 for a subscriber session. 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.

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)*.
- Configure the SRC Diameter application. See "Configuring the Diameter Application (SRC CLI)" on page 70.
- Configure the NAS groups. See "Configuring the NAS Groups (SRC CLI)" on page 78.
- Configure the SAE to manage AAA devices. See "Configuring the SAE to Manage AAA Devices" on page 82.
- Configure AAA policies. See "Configuring AAA Policies (SRC CLI)" on page 84.

Related Documentation

- SIC Dynamic Authorization Support Overview on page 12
- How the Dynamic Authorization Process Works in the SIC on page 14
- *Subscriber Information Collector Overview*

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.

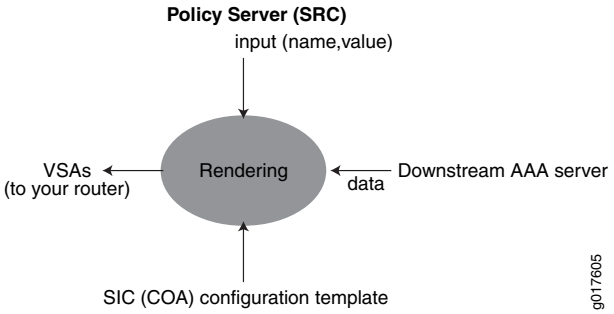
The SIC dynamic authorization function includes:

- 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 1 on page 13.

Figure 1: 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 14
- Managing Dynamic Services on page 11

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;
        }
      }
    }
  }
}
```

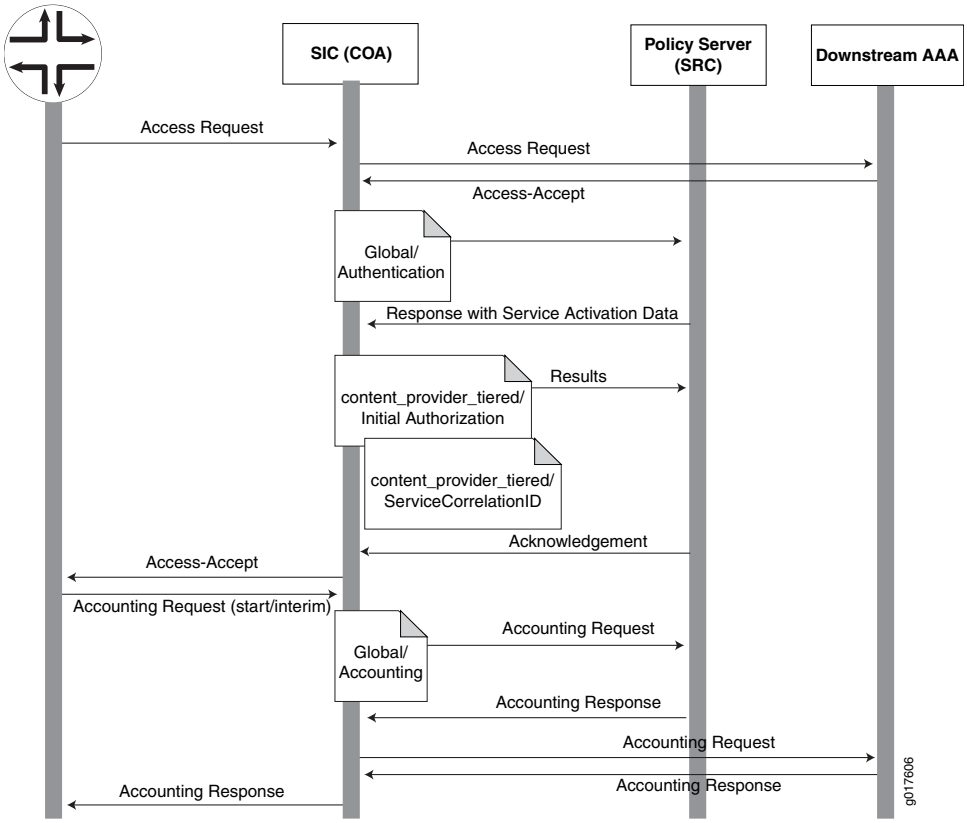
The SIC renders the Access-Accept message, then informs the SAE about rendering successes and failures in another request. The SAE sends an acknowledgement back to the SIC, which in turn sends a rendered Access-Accept message to the NAS.

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 2 on page 16 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 2: Initial Authorization and Accounting Timing Sequence



Service Activation and Deactivation

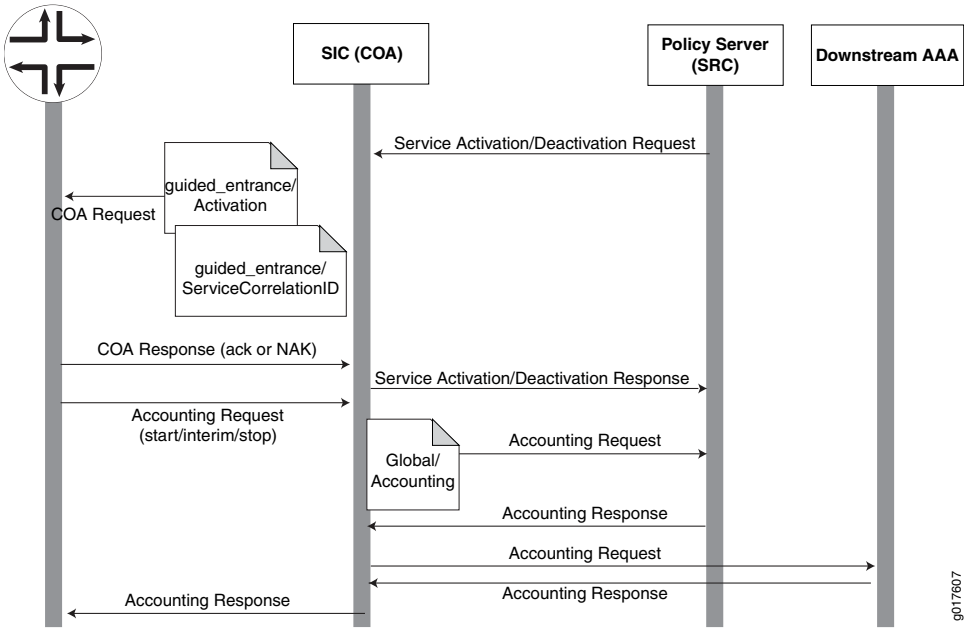
This section describes the service activation and deactivation scenarios. The sequences for activation and deactivation are identical except that the activation sequence uses activation requests and the deactivation sequence uses deactivation requests.

A service activation begins with an activation request from the SAE to the SIC. Figure 3 on page 17 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 15.

Figure 3 on page 17 shows the activation and deactivation timing sequences. For purposes of this illustration, the SIC and SRC are shown in two distinct rectangles.

Figure 3: Activation and Deactivation Timing Sequences



Abort Session Requests

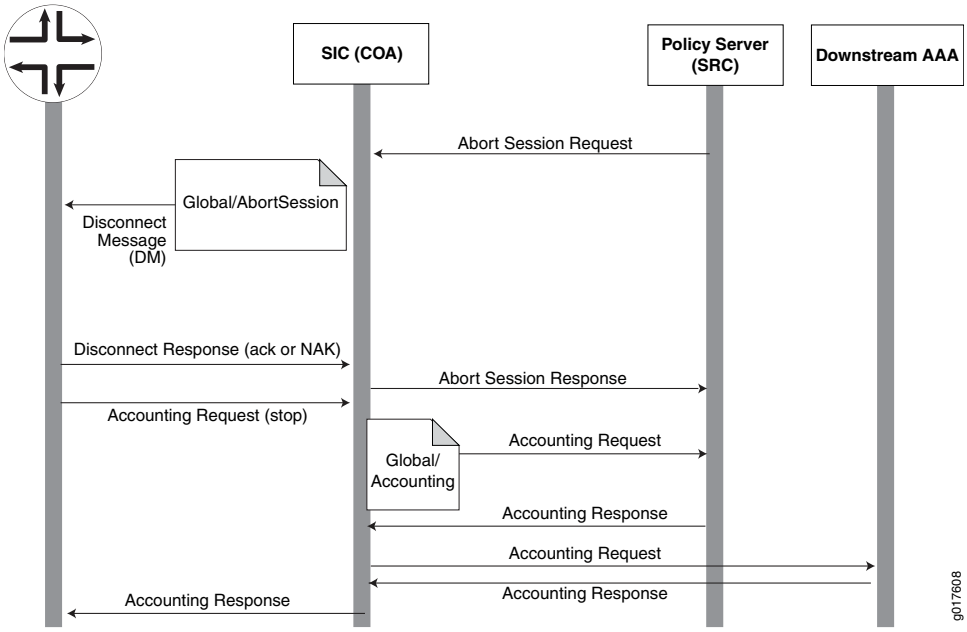
If the SAE receives an abort session request, it sends it to the SIC. The SIC, using the global service template and Abort-Session mode, renders the request and sends it, along with a service correlation ID, as a Disconnect Message (DM) to the NAS. The NAS responds with an ack or NAK. The SIC then sends 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 15.

Figure 4 on page 18 shows the abort session timing sequence. For purposes of this illustration, the SIC and SRC are shown in two distinct rectangles.

Figure 4: Abort Session Timing Sequence



Related Documentation

- SIC Dynamic Authorization Support Overview on page 12
- Device and Service Template Configuration Overview (SRC CLI) on page 29
- Dynamic Authorization Targets (SRC CLI) on page 18

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 29
- SIC Dynamic Authorization Support Overview on page 12
- How the Dynamic Authorization Process Works in the SIC on page 14

PART 2

Configuration

CHAPTER 4

Configuration Tasks for CoA Script Services

Configuring COA Script Services

To support COA message exchange in an SRC network, configure a script service that can be activated on a third-party device. The script service defines the parameters needed to activate or deactivate services for a subscriber session, such as the address of the third-party device. This script service is activated for the subscriber session whose services are activated or deactivated. For detailed information about configuring script services, see *Customizing Service Implementations*.

When you use the COA script service with third-party devices that do not notify the SAE about subscriber events, you must set up the Monitoring Agent application to handle RADIUS accounting request packets.

For information about configuring services on the third-party device, see the device's software documentation.

The tasks to set up the SRC software for COA message exchange are:

- "Configuring Monitoring Agent to Receive RADIUS Accounting Messages" on page 22
- "Creating the COA Script Service (SRC CLI)" on page 22
- "Configuring the COA Script Service (SRC CLI)" on page 23
- "Configuring Subscriptions to the COA Script Service" on page 24

The SRC software includes a sample script service that you can configure to exchange COA messages with the third-party device. You can use the sample service definition

and customize it for your environment by modifying the service substitutions. For information about the sample COA script service, see "Example: Using the Sample COA Script Service" on page 27.

Related Documentation

- COA Script Service Overview on page 9
- Defining RADIUS Attributes for COA Requests with the API on page 24
- *Setting Up Script Services*
- Parameters for Sample COA Script Service on page 9

Configuring Monitoring Agent to Receive RADIUS Accounting Messages

If you install the Monitoring Agent application on the same host as the RADIUS server, you must disable the MonAgent.radius.server property.

You can configure Monitoring Agent to act as a pseudo–RADIUS server that listens for RADIUS accounting packets sent to the RADIUS accounting port. To receive RADIUS packets from RADIUS clients:

- Make sure there is no other RADIUS server listening on the RADIUS accounting port, and enable the MonAgent.radius.server property.
- Configure the shared secret between the RADIUS server and the RADIUS client by specifying the MonAgent.radius.secret.<IP address> property.

For information about installing and using Monitoring Agent, see the *SRC Sample Applications Guide*.

Related Documentation

- Configuring the COA Script Service (SRC CLI) on page 23
- Defining RADIUS Attributes for COA Requests with the API on page 24

Creating the COA Script Service (SRC CLI)

To create the script service:

1. From configuration mode, enter the service configuration. In this sample procedure, the service is configured in the global service scope, and COAservice is the name of the service.

   user@host# edit services global service COAservice

2. Configure the type of service.

   [edit services global service COAservice]
   user@host# set type script

3. (Optional) Specify whether the service is visible only to administrators who have permission to see secret information.

   [edit services global service COAservice]

```
user@host# set secret
```

4. Configure URL as the type of script that the sample COA script service uses.

```
[edit services global service COAservice]
user@host# set script script-type url
```

5. Configure net.juniper.smgt.sae.coa.CoaService as the name of the class that implements the script service.

```
[edit services global service COAservice]
user@host# set script class-name net.juniper.smgt.sae.coa.CoaService
```

6. Configure the URL of the script service or the path and filename of the service. Copy the */lib/coa.jar* file used by the script service to a location that is accessible by a URL (such as an FTP or HTTP server). In this sample procedure, the *coa.jar* file was copied to the */opt/UMC/sae/var/run* directory.

```
[edit services global service COAservice]
user@host# set file file:///opt/UMC/sae/var/run/coa.jar
```

7. (Optional) Verify your configuration.

```
[edit services global service COAservice]
user@host# show
type script;
status active;
available;
script {
  script-type url;
  class-name net.juniper.smgt.sae.coa.CoaService;
  file file:///opt/UMC/sae/var/run/coa.jar;
}
```

After you create the script service, you need to configure parameters for the script service. For more information about configuring script services and parameters, see *SRC Script Services Overview*.

Related Documentation

- COA Script Service Overview on page 9
- Configuring Subscriptions to the COA Script Service on page 24
- Configuring COA Script Services on page 21
- Configuring the COA Script Service (SRC CLI) on page 23
- Parameters for Sample COA Script Service on page 9

Configuring the COA Script Service (SRC CLI)

To configure the script service, you provide parameter substitutions with the values that are in the service definitions.

To configure parameters:

1. From configuration mode, enter the service parameter configuration. In this sample procedure, the service called COAservice is configured in the global service scope.

   user@host# edit services global service COAservice parameter

2. (Optional) Configure actual values for other parameters.

   [edit services global service COAservice parameter]
   user@host# set substitution [*substitution...*]

   The script file */ SDK/scriptServices/coa/ldif/BOD1M.ldif* in the **SDK+AppSupport+Demos+Samples.tar.gz** file provides parameters specified by the sample COA script service. You can use the sample script service as a starting point. See "Parameters for Sample COA Script Service" on page 9.

Related Documentation

- COA Script Service Overview on page 9
- Configuring Subscriptions to the COA Script Service on page 24
- Creating the COA Script Service (SRC CLI) on page 22
- Configuring COA Script Services on page 21
- Example: Using the Sample COA Script Service on page 27

Configuring Subscriptions to the COA Script Service

You need to configure subscriptions to the COA script service. You can set up the subscriptions to activate immediately on login.

For more information, see *Adding Subscribers (SRC CLI)*.

Related Documentation

- COA Script Service Overview on page 9
- Configuring COA Script Services on page 21
- Configuring the COA Script Service (SRC CLI) on page 23
- Example: Using the Sample COA Script Service on page 27

Defining RADIUS Attributes for COA Requests with the API

The SRC software provides two ways to define RADIUS attributes for dynamic RADIUS authorization requests:

- Service definition (see "Configuring the COA Script Service (SRC CLI)" on page 23)
- SAE core API

NOTE: **Parameters set in the API override parameters set by the service definition.**

To send dynamic RADIUS authorization requests with the SAE core API, the script service uses the sendDynamicRadius and getRouterDynRadiusAddr methods in the ServiceSessionInfo interface to provide the content of the RADIUS packet for the dynamic authorization request to the router that is attached to the service session.

For information about the ServiceSessionInfo interface, see the script service documentation in the SAE core API documentation on the Juniper Networks Web site at http://www.juniper.net/techpubs/software/management/src/api-index.html.

For a sample implementation, see the following file in the **SDK+AppSupport+Demos+Samples.tar.gz**file:

SDK/scriptServices/coa/java/net/juniper/smgt/scriptServices/coa/CoaService.java.

Related Documentation

- COA Script Service Overview on page 9
- Configuring COA Script Services on page 21
- Creating the COA Script Service (SRC CLI) on page 22
- Configuring Monitoring Agent to Receive RADIUS Accounting Messages on page 22

CHAPTER 5

Example

Example: Using the Sample COA Script Service

To use the sample COA script service provided:

1. Import the sample script service using an LDAP browser.

   The */SDK/scriptServices/coa/ldif/BOD1M.ldif* file (in the **SDK+AppSupport+Demos+Samples.tar.gz** file) is the sample service definition for exchanging COA messages with a Cisco 10000 Series router.

2. Copy the */lib/coa.jar* file used by the script service to a location that is accessible to the SAE by a URL, such as an FTP or HTTP server. If you do not have multiple SAEs, it can be convenient to copy the file to the */var/run* directory in the SAE installation directory (*/opt/UMC/sae* by default).

3. Modify the service substitutions for your device.

   You can make these substitutions by defining the parameter substitutions in the BOD1M service with the SRC CLI or by passing the values through the SAE core API.

   For information about parameter substitutions, see "Configuring the COA Script Service (SRC CLI)" on page 23. For information about passing the values through the SAE core API, see "Defining RADIUS Attributes for COA Requests with the API" on page 24.

4. Configure a subscription to the BOD1M service that is activated on login.

   For more information about subscriptions, see *Subscriptions Overview*.

If you are modifying the sample application, add the *sae.jar* and *logger.jar* files to the classpath when you compile your application. These two files can be found in the *lib* directory of the SAE installation directory.

Related Documentation

- COA Script Service Overview on page 9
- Configuring Subscriptions to the COA Script Service on page 24
- Configuring COA Script Services on page 21
- Creating the COA Script Service (SRC CLI) on page 22

CHAPTER 6

Configuration Tasks for Dynamic Authorization

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 6 on page 30 describes the available capabilities and associated values.

Table 6: 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 activatin authorization or activation. **Access-Accept**—Indicates that the router supports activating services only 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 temple. You must use the provided modes; you cannot create new modes.

Table 7 on page 31 lists the modes and attributes for global service templates.

Table 7: 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 31.

Table 8 on page 31 lists the modes and attributes for global service templates.

Table 8: Global Service Template Modes

Mode	Description
Authentication	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. Unlike modes in service templates, this mode renders requests to the SRC software and not to the router.
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 9 on page 32 lists the attributes, explains their parameters, and describes their behavior.

Table 9: Attributes for All Modes

Attribute	Description
required	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. Options • **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 9: Attributes for All Modes *(continued)*

Attribute	Description
override	Whether or not the renderer finds the attribute in the downstream AAA server response, it creates the attribute name with the specified value. Options • **name** ***name*** —Name of the attribute. The name must match a defined RADIUS attribute in the downstream AAA server response. • **value** ***value*** —Set the attribute to this value.
default	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. Options • **name** ***name*** —Name of the attribute. The name must match a defined RADIUS attribute in the downstream AAA server response. • **value** ***value*** —Set the attribute to this value. • **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.
normal	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-attribute*, the rendering does not fail in this case. Options • **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 9: Attributes for All Modes *(continued)*

Attribute	Description
parameterized	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. Options • **name** *name*—Name of the attribute. The specified name must match a defined RADIUS attribute in the downstream AAA server response. • **format** *format*—In a form of "$(p1) $(p2) ... $(pn) [$p(n+1)]". Behaves much like sprintf in C; you can intersperse literal text in between parameter definitions. Unlike sprintf, format 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: **\$, \', \", \[, \], \(, \).**

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 10 on page 34.

Table 10: 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 9 on page 32 are also support for tagged attribute configurations.

Related Documentation

- Managing Dynamic Services on page 11
- SIC Dynamic Authorization Support Overview on page 12
- How the Dynamic Authorization Process Works in the SIC on page 14

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 29.

- Configure the device template used by the SIC group.

  See "Configuring Device Templates (SRC CLI)" on page 36.

- Configure the device capabilities.

  See "Configuring the Device Capabilities Supported in the Device Template (SRC CLI)" on page 37.

- Configure the service template.

  See "Configuring SIC Service Templates (SRC CLI)" on page 39.

- Configure any tagged attributes for the service template.

See "Configuring Tagged Attributes in SIC Service Templates (SRC CLI)" on page 48.

- Configure the global service template.

  See "Configuring Global Service Templates (SRC CLI)" on page 56.

Related Documentation

- Managing Dynamic Services on page 11
- SIC Dynamic Authorization Support Overview on page 12
- How the Dynamic Authorization Process Works in the SIC on page 14

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)* and *Configuring Dictionaries for the SIC Group (SRC CLI)*.

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)*
- Configuring the Device Capabilities Supported in the Device Template (SRC CLI) on page 37
- Device and Service Template Configuration Overview (SRC CLI) on page 29
- *Sample Service Templates*

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 11 on page 37 describes the available capabilities and associated values.

Table 11: 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.
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 36
- Device and Service Template Configuration Overview (SRC CLI) on page 29
- *Sample Service Templates*

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 29
- Configuring the Device Capabilities Supported in the Device Template (SRC CLI) on page 37
- Configuring Tagged Attributes in SIC Service Templates (SRC CLI) on page 48
- Configuring Global Service Templates (SRC CLI) on page 56

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 29 for details on configuring the options in the following procedure.

- Creating an SIC Service Template (SRC CLI) on page 40
- Configuring the Mode of the SIC Service Template (SRC CLI) on page 40
- Configuring Variables for the SIC Service Template (SRC CLI) on page 40
- Configuring Normal Attributes for the SIC Service Template (SRC CLI) on page 41
- Configuring Required Attributes for the SIC Service Template (SRC CLI) on page 42

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
   ```

4. 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
   ```

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.

6. 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:

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 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 a 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 36
- Device and Service Template Configuration Overview (SRC CLI) on page 29
- Configuring the Device Capabilities Supported in the Device Template (SRC CLI) on page 37
- Configuring Tagged Attributes in SIC Service Templates (SRC CLI) on page 48
- Configuring Global Service Templates (SRC CLI) on page 56

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 29
- Configuring the Device Capabilities Supported in the Device Template (SRC CLI) on page 37
- Configuring Tagged Attributes in SIC Service Templates (SRC CLI) on page 48
- Configuring Global Service Templates (SRC CLI) on page 56

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 49
- Configuring Normal Attributes in a Tagged Attribute Group (SRC CLI) on page 49
- Configuring Default Attributes in a Tagged Attribute Group (SRC CLI) on page 50

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 a 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 29
- Configuring the Device Capabilities Supported in the Device Template (SRC CLI) on page 37
- Configuring Global Service Templates (SRC CLI) on page 56

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 29
- Configuring the Device Capabilities Supported in the Device Template (SRC CLI) on page 37
- Configuring Tagged Attributes in SIC Service Templates (SRC CLI) on page 48
- Configuring Global Service Templates (SRC CLI) on page 56

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 56
- Configuring the Mode of the SIC Global Service Template (SRC CLI) on page 57
- Configuring Variables for the SIC Global Service Template (SRC CLI) on page 57
- Configuring Normal Attributes for the SIC Global Service Template (SRC CLI) on page 58
- Configuring Required Attributes for the SIC Global Service Template (SRC CLI) on page 59
- Configuring Default Attributes for the SIC Global Service Template (SRC CLI) on page 60
- Configuring Parameterized Attributes for the SIC Global Service Template (SRC CLI) on page 61
- Configuring Override Attributes for the SIC Global Service Template (SRC CLI) on page 62

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 a 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 29
- Configuring the Device Capabilities Supported in the Device Template (SRC CLI) on page 37
- Configuring Tagged Attributes in SIC Service Templates (SRC CLI) on page 48

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 78.

2. Configure the SAE to manage SAE devices.

   See "Configuring the SAE to Manage AAA Devices" on page 82.

3. Configure the AAA policy rules.

   See "Configuring AAA Policies (SRC CLI)" on page 84.

Related Documentation

- Managing Dynamic Services on page 11
- SIC Dynamic Authorization Support Overview on page 12
- How the Dynamic Authorization Process Works in the SIC on page 14

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 12
- *RADIUS Authentication/Authorization and Accounting Data Flow*
- *RADIUS and Diameter Configuration for the SIC Overview (SRC CLI)*
- How the Dynamic Authorization Process Works in the SIC on page 14

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 66.

2. Configure the Diameter application.

See "Configuring the Diameter Application (SRC CLI)" on page 70.

3. Configure the SRC Diameter server.

   See "Configuring Diameter Peers (SRC CLI)" on page 76.

Related Documentation

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

Configuring the SIC Diameter Server (SRC CLI)

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 70 and "Configuring Diameter Peers (SRC CLI)" on page 76.

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 70
- Configuring the Diameter Client Properties on page 74
- Configuring the Diameter Server Properties on page 74
- Configuring Logging Destinations on page 75

Configuring the Diameter Application Properties

The SRC software supports Diameter application properties such as Juniper Networks Session Resource Control (JSRC), Packet-Triggered Subscribers and Policy Control

(PTSP), and GX-Plus. JSRC and PTSP communicates with the Service Activation Engine (SAE) (remote SRC peer), whereas GX-Plus communicates with the Policy and Charging Rules Function (PCRF).

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. For PTSP, realm-based Diameter routing is not used.

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.

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*.
- To manage policies for PTSP peers on MX Series routers, see *Configuring PTSP to Manage Subscriber-Level Policies*.

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;
}
```

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. 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 70
- *Viewing SRC Diameter Server State (SRC CLI)*

Configuring the NAS Groups (SRC CLI)

Tasks to configure the NAS groups are:

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

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.

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 page 11

PART 3

Index

Index

T