JUNIPER NETWORKS

SRC PE Software

Solutions Guide

Release

4.0.x

Published: 2010-06-04

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

SRC PE Software Solutions Guide
Release 4.0.x

Writing: Linda Creed, Justine Kangas, Betty Lew
Editing: Fran Mues
Illustration: Nathaniel Woodward
Cover Design: Edmonds Design

Revision History
May 2010—Revision 1

The information in this document is current as of the date listed in the revision history.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The 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.

SOFTWARE LICENSE

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

For complete product documentation, please see the Juniper Networks Web site at www.juniper.net/techpubs.

END USER LICENSE AGREEMENT

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.

e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the

Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at http://www.gnu.org/licenses/gpl.html, and a copy of the LGPL at http://www.gnu.org/licenses/lgpl.html .

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentés confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattaché, soient redigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Abbreviated Table of Contents

Table of Contents

List of Figures

List of Tables

About the Documentation

SRC Documentation and Release Notes

For a list of related SRC documentation, see http://www.juniper.net/techpubs/.

If the information in the latest *SRC Release Notes* differs from the information in the SRC guides, follow the *SRC Release Notes*.

Audience

This documentation is intended for experienced system and network specialists working with routers running JUNOS® and JUNOSe Software in an Internet access environment. We assume that readers know how to use the routers, directories, and RADIUS servers that they will deploy in their SRC networks. If you are using the SRC software in a cable network environment, we assume that you are familiar with the PacketCable Multimedia Specification (PCMM) as defined by Cable Television Laboratories, Inc. (CableLabs) and with the Data-over-Cable Service Interface Specifications (DOCSIS) 1.1 protocol. We also assume that you are familiar with operating a multiple service operator (MSO) multimedia-managed IP network.

Documentation Conventions

Table 1 on page xxii defines the notice icons used in this guide. Table 2 on page xxii defines text conventions used throughout this documentation.

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.

Table 2: 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**.
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/src/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.

Table 2: Text Conventions *(continued)*

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

Providing Specialized Services in an SRC Environment

CHAPTER 1

Managing Tiered and Premium Services with QoS on JUNOSe Routers (SRC CLI)

Overview of QoS on JUNOSe Routers

Tiered Internet access and premium services such as video on demand, gaming, or videoconferencing require QoS profiles to be running on the subscriber interface on the router running JUNOSe Software. The router allows only one QoS profile to be attached to an interface at one time. Therefore, as a subscriber activates and deactivates different services, the QoS profile running on the interface needs to change. Also, as subscribers activate services, they may have multiple QoS services running at the same time; for example, internet-gold with videoconferencing.

With the SRC software, you can:

- Dynamically manage QoS profiles on the router running JUNOSe Software to control a combination of services that require QoS.
- Update the directory with a list of QoS profiles that are currently configured on a router running JUNOSe Software.
- Search the directory for QoS policy information.

Related Topics

- Dynamically Managing QoS Profiles on page 4
- Delivering QoS Services in a Cable Environment
- Configuring QoS Profile-Tracking Plug-Ins (SRC CLI) on page 9
- Updating QoS Profile Data in the Directory on page 12
- Examples: Searching for QoS Information on page 13

Dynamically Managing QoS Profiles

The SAE provides a QoS-tracking plug-in (QTP) that you can use to ensure that, as a subscriber activates and deactivates services, the required QoS profile is attached to the subscriber interface. With the QTP, the QoS profile selected is based on the activation state of an aggregation of services, not just one service.

For example, a subscriber activates a QoS service on a subscriber interface that requires a QoS profile that supports 512 best effort. The subscriber then activates a faster service (for example, 1024 best effort), as well as video on demand, and now has two QoS services running on an interface. The subscriber now needs a QoS profile to be attached to the interface that supports both video on demand and 1024 best-effort service. The QTP can determine which QoS profile the subscriber needs, and can cause the existing QoS profile to be removed from the subscriber interface and the new QoS profile to be attached to the interface.

Note that if a profile is installed on a subscriber interface and the QTP installs a new profile, the new profile is based on QoS services that are currently active. The new profile does not combine the functionality of the previous profile with the new profile. For example, if a subscriber has a default policy with QoS profile be-512 installed on the subscriber interface, and the subscriber activates a video-on-demand service, the QTP does not combine the functionality of be-512 with the profile that supports video on demand.

How QoS Profile Tracking Works

The SAE manages policies on router interfaces through service sessions. Service session configurations contain the policy that needs to be installed on an interface when a service is activated. The policy definition can include the name of a QoS profile to attach to the interface when the policy is installed.

When you set up the QTP, you create a QoS profile attachment service. The purpose of this service is to attach the required QoS profile to an interface. This service is hidden from subscribers and is under only QTP control.

Because profiles need to be changed only when QoS services are activated or deactivated, the QTP tracks services and reacts to service state changes by adjusting the QoS profile attachment as needed by deactivating and activating the QoS profile attachment service.

Subscribers who need their services managed by the QTP are subscribed to the QoS profile attachment service.

Identifying QoS Services

When you set up a service, you identify the service as a QoS service in one of the fields in the service definition. For example, you can assign a service name or category to indicate that the service is a QoS service, or you could assign the QTP instance name in the Tracking Plugin field.

When the SAE notifies the QTP that a service has been activated or deactivated, the QTP determines whether it is a QoS service by searching attributes in the service object.

The QTP uses a search filter that you set up to search an attribute for the information that you assigned to the service to indicate that it is a QoS service.

For example, suppose you enter myqtp in the tracking plug-in field of QoS services to indicate that the service is a QoS service. You would set up the search filter to search tracking plug-in attributes for any service that contains myqtp:

```
(attribute.trackPlug=*myqtp*)
```

Or you might configure the category to indicate that a service is a QoS service. The following filter searches service category attributes for any entry that contains ultra, video on demand, or video telephony:

```
(|(serviceCategory=*ultra*)(|(serviceCategory=*video on
  demand*)(serviceCategory=*video telephony*)))
```

To obtain a list of attribute names for the sspService object class, see the LDAP schema documentation in SDK+AppSupport+Demos+Samples.tar.gz file in the folder *SDK/doc/ldap* or on the Juniper Networks Web site at

http://www.juniper.net/techpubs/software/management/src

Determining the QoS Profile

After the QTP determines that a service is a QoS service, it needs to obtain the name of the QoS profile for the service. The QTP generates a QoS profile name based on active QoS services as follows:

1. Obtains QoS profile input values.

   The QTP obtains these values by taking the value of an attribute in the service definition. You specify which attribute that you want the QTP to use as the input value. For example, you can specify the service name, the category, or the contents of the design and graphics attribute.

2. Compiles a list of the QoS profile input values.
3. Removes duplicate values from the list.
4. Sorts the remaining list by using a case-sensitive alphanumeric comparison.
5. Concatenates the values with a separator. The default value for the separator is a hyphen (-). You can specify a different separator.

   Table 3 on page 5 shows how lists of QoS profile input values are sorted and then concatenated.

Table 3: Examples of Concatenated QoS Profile Input Values

Input – QoS Profile Input Values	Output – Concatenated Name
be512, vod	be512-vod
game, be1024, vod	be1024-game-vod

Table 3: Examples of Concatenated QoS Profile Input Values *(continued)*

Input – QoS Profile Input Values	Output – Concatenated Name
be128	be128

6. Adds a prefix to the resulting name. The default prefix is qos-profile. (You can specify a different value.) The output from our examples now looks like this:
   - qos-profile-be512-vod
   - qos-profile-be1024-game-vod
   - qos-profile-be128

The names that result from this process are the QoS profile names.

As you can see from this process, you need to design services and configure the QTP so that the resulting QoS profile names match the names of the QoS profiles configured on the router running JUNOSe Software.

Typically, a QoS designer creates a number of QoS profiles that support all the services that are expected to be used. This design results in various QoS profiles that need to be configured on each router. If a required QoS profile is not configured on the router, the hidden QoS profile attachment service cannot be activated. Services are still activated for the subscriber, but the services will not provide the expected traffic requirements. When this happens, the SAE logs the error but does not send an error message to the subscriber.

Setting Up Policy Groups

You need to create two types of policy groups in your QTP configuration. The QoS profile attachment service needs a policy group that attaches the required QoS profile to the subscriber interface when the attachment service is activated. QoS services need policy groups that classify traffic and specify the action to take on traffic that matches the classifier. (You can set up traffic classifiers to match any traffic.)

Policy Group for QoS Profile Attachment Service

The policy group for the hidden QoS profile attachment service must have an egress policy list with only one policy rule that contains a QoS profile attachment action. The QoS profile attachment action must have a variable parameter in the QoS profile field.

NOTE: The policy group for the QoS profile attachment service must contain only one egress policy list and must contain one and only one QoS profile attachment action. Otherwise, the SRC software will require a license for the hidden service.

When the profile attachment service is activated, the QTP substitutes the QoS profile attribute in the policy with the QoS profile name that it determined. The service then loads the policy.

The following example creates a policy group for the QoS profile attachment service. This policy group does not match any traffic.

1. Create a policy group called Pg-qos-attach, and add an egress policy list.
2. In the egress policy list, create a policy rule that has a QoS profile attachment action with QoS profile qpName.

   By default, the QTP looks for qpName as the variable parameter.

   When the QTP determines the required QoS profile name, it substitutes qpName with the value that it acquired.

Setting Up Services

You need to set up a QoS profile attachment service and QoS services. Both types of services are value-added (SSP) services.

In the QoS profile attachment service, assign the policy group that you configured for the service. For example, policyGroupName=Pg-qos-attach, ou=ent, o=Policies, o=umc.

In QoS services, assign the policy group that you configured for the service.

Subscribe subscribers to the QoS profile attachment service and to the appropriate QoS services.

Reestablishing Default QoS Profile

A default QoS profile may be installed on the subscriber interface before the QTP installs QoS profiles in response to the activation of QoS services. For example, a profile may have been attached to the subscriber interface when the default policy was installed. Once QoS services are no longer active on the interface, the QTP can reestablish the QoS profile that was installed on the interface before the QTP began tracking services and installing profiles on the interface.

Example: How QTP Activates a QoS Service

The following example shows the process that QTP uses when a subscriber activates a QoS service. In this example, QoS profile input values are taken from the service name attribute. The hidden QoS profile attachment service is named svc-qos-attach. The svc-qos-attach service contains a policy that has the variable parameter qpName assigned as the QoS profile name.

1. The subscriber does not have any active services.
2. The subscriber activates service be512, which is a QoS service.
   a. The SAE sends a Service Session Start event to the QTP.
   b. The QTP searches an attribute in the service definition and determines that the service is a QoS service.
   c. Using the SAE Common Object Request Broker Architecture (CORBA) remote application programming interface (API), the QTP gets a list of the subscriber's active QoS services.

The list contains only service be512 because that is the only service that the subscriber has activated.

d. The QTP adds the default prefix to the QoS profile input value to obtain the QoS profile name. The result is:

   qos-profile-be512

e. The QTP deactivates the hidden svc-qos-attach service. Because this svc-qos-attach service was not active before, this operation does not have any effect.

f. The QTP activates the hidden svc-qos-attach service, and it substitutes variable parameter qpName with $'qos-profile-be512' as the QoS profile name in the policy.

g. The policy loads qos-profile-be512 on the subscriber interface.

3. The subscriber activates service vod, which is a QoS service.

   a. The SAE sends a Service Session Start event to the QTP.

   b. QTP searches attributes in active service definitions and determines that the service is a QoS service.

   c. The QTP gets a list of the subscriber's active QoS services. The result is:

      be512, vod

   d. The QTP sorts the list and concatenates the QoS profile input values with the separator. The result is:

      be512-vod

   e. The QTP adds the default prefix to the concatenated name to obtain the QoS profile name. The result is:

      qos-profile-be512-vod.

   f. The QTP deactivates the hidden svc-qos-attach service.

   g. The QTP activates the hidden svc-qos-attach service, and it substitutes variable parameter qpName with $'qos-profile-be512-vod' as the QoS profile name in the policy.

   h. The policy loads qos-profile-be512-vod.

4. The subscriber deactivates service vod.

   a. The QTP follows the same procedure as in Step 2 above and determines that the QoS profile name is qos-profile-vod.

   b. The QTP deactivates the hidden svc-qos-attach service.

c. The QTP reactivates the hidden svc-qos-attach service, and it substitutes variable parameter qpName with $'qos-profile-be512' as the QoS profile name in the policy.

d. The policy loads qos-profile-be512.

Related Topics

- Overview of QoS on JUNOSe Routers on page 3
- Configuring QoS Profile-Tracking Plug-Ins (SRC CLI) on page 9
- Configuring QoS Profile Attachment Actions (SRC CLI)
- Configuring Search Filters for QoS Profile-Tracking Plug-Ins on page 11
- Updating QoS Profile Data in the Directory on page 12

Configuring QoS Profile-Tracking Plug-Ins (SRC CLI)

Use the following configuration statements to configure the QoS profile tracking plug-in with the SRC CLI:

```
shared sae configuration plug-ins name name qos-profile-tracking {
  threads threads ;
    default-qos-profile default-qos-profile ;
    separator separator ;
    qos-profile-prefix qos-profile-prefix ;
    service-selection-attribute service-selection-attribute ;
    search-filter search-filter ;
    invisible-qos-service invisible-qos-service ;
    qos-profile-parameter-name qos-profile-parameter-name ;
}
```

1. From configuration mode for the QoS profile tracking plug-in.

   ```
   user@host# edit shared sae configuration plug-ins name QosTracking
     qos-profile-tracking
   ```

2. Configure the number of working threads that all QTP instances share when they process QTP events.

   ```
   [edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
   user@host# set threads threads
   ```

3. Configure the name of the QoS profile that is attached to the interface when QoS services have been deactivated.

   See "Dynamically Managing QoS Profiles" on page 4.

   ```
   [edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
   user@host# set default-qos-profile default-qos-profile
   ```

4. Configure the character that is placed between QoS profile input values when the system concatenates the values during the process of creating QoS profile names.

   ```
   [edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
   ```

```
user@host# set separator separator
```

5. Configure the prefix added to the QoS service name as part of the process to determine the name of the QoS profile that needs to be attached to an interface for a particular service.

```
[edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
user@host# set qos-profile-prefix qos-profile-prefix
```

6. Configure the name of the attribute in the service definition that you want the QTP to use as QoS profile input values.

```
[edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
user@host# set service-selection-attribute service-selection-attribute
```

7. Configure the search filter that the SAE uses to search service objects in the directory to find QoS services.

See "Configuring Search Filters for QoS Profile-Tracking Plug-Ins" on page 11

```
[edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
user@host# set search-filter search-filter
```

8. Configure the name of the hidden QoS profile attachment service that the QTP uses to attach QoS profiles to and remove QoS profiles from a router interface.

```
[edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
user@host# set invisible-qos-service invisible-qos-service
```

9. Configure the name of the variable parameter used in the QoS profile name field in the QoS profile attachment action of the policy group that is assigned to the hidden QoS service.

```
[edit shared sae configuration plug-ins name QosTracking qos-profile-tracking]
user@host# set qos-profile-parameter-name qos-profile-parameter-name
```

10. Verify your configuration.

```
[edit shared sae configuration plug-ins name QosTracking
qos-profile-tracking]
user@host# show
threads 1;
default-qos-profile ;
separator -;
qos-profile-prefix qos-profile;
service-selection-attribute serviceName;
search-filter (attribute.trackPlug=);
invisible-qos-service svc-qos-attach;
qos-profile-parameter-name qpName;
```

Related Topics

- Updating QoS Profile Data in the Directory on page 12
- Query Fields on page 12
- Examples: Searching for QoS Information on page 13

- Overview of QoS on JUNOSe Routers on page 3

Configuring Search Filters for QoS Profile-Tracking Plug-Ins

The SAE uses a search filter to search service objects in the directory to find QoS services. You can set up the filter to search the values of any attribute in the service object, such as service name, category, or tracking plug-in. The search is successful when a value matches the filter.

To configure the search:

- Create a filter in a format similar to the LDAP search filter. Table 4 on page 11 lists the values that you can use for filters. Each filter string <filter> contains a simplified LDAP query.

Table 4: Settings for Filter Strings

Filter String	Action
()	Matches no objects
(*)	Matches all objects
List of <attribute>= <value> pairs <attribute>—Name of a property or attribute <ldapAttributeName> <value>—One of the following • * (asterisk) • Explicit string • String that contains an * **Note:** To define a special character (* & , ! | \ )in a string, precede it with the backslash symbol (\).	• If <value> is *, checks for any value. • If <value> is an explicit string, checks whether any value of the property matches the string, regardless of case. • If <value> is a string that contains a *, checks whether any value of the property contains the string, regardless of case.
(&<filter><filter>...)	True if all filters match
(|<filter><filter>...)	True if at least one filter matches
(!<filter>)	True if the filter does not match

The default is attribute.trackPlug=; note that you need to add a search value after the equal sign. For example:

- To search tracking plug-in attributes for any entry that contains qtp:

(attribute.trackPlug=*qtp*)

- To search service category attributes for any entry that contains ultra, video on demand, or video telephony:

(|(serviceCategory=*ultra*)(|(serviceCategory=*video on demand*)(serviceCategory=*video telephony*)))

Related Topics

- Dynamically Managing QoS Profiles on page 4
- Configuring QoS Profile-Tracking Plug-Ins (SRC CLI) on page 9
- Updating QoS Profile Data in the Directory on page 12
- For information about obtaining a list of attribute names for the sspService object class, see the documentation for the LDAP schema in SDK+AppSupport+Demos+Samples.tar.gz file in the folder *SDK/doc/ldap* or on the Juniper Networks Web site at

  http://www.juniper.net/techpubs/software/management/src
- Examples: Searching for QoS Information on page 13

Updating QoS Profile Data in the Directory

You can update the directory with a list of QoS profiles that are currently configured on a router running JUNOSe Software.

Related Topics

- Dynamically Managing QoS Profiles on page 4
- Configuring QoS Profile-Tracking Plug-Ins (SRC CLI) on page 9
- Configuring Search Filters for QoS Profile-Tracking Plug-Ins on page 11
- Query Fields on page 12
- Overview of QoS on JUNOSe Routers on page 3

Query Fields

The following fields appear in the Query dialog box of the Policy Editor.

Condition Type

- Object to be searched.
- Value—router, QoS profile, or policy group
- Default—No value

Condition Value

- Name of the QoS profile, router, or policy group that you want to search.
- Value—Name of the router, QoS profile, or policy group. If you selected router or policy group as a condition type, you can select a name from the drop-down menu. If the condition type is QoS profile, continue selecting entries in the drop-down menu until you reach the name of a policy group.
- Default—No value

Find

- Object that you want to find. The software searches for this object on the QoS profile, router, or policy group defined in condition type and condition value.
- Value—Name of the router, QoS profile, or policy group. If you selected router or policy group as a condition type, you can select a name from the drop-down menu. If the condition type is QoS profile, continue selecting entries in the drop-down menu until you reach the name of a policy group.
- Default—No value

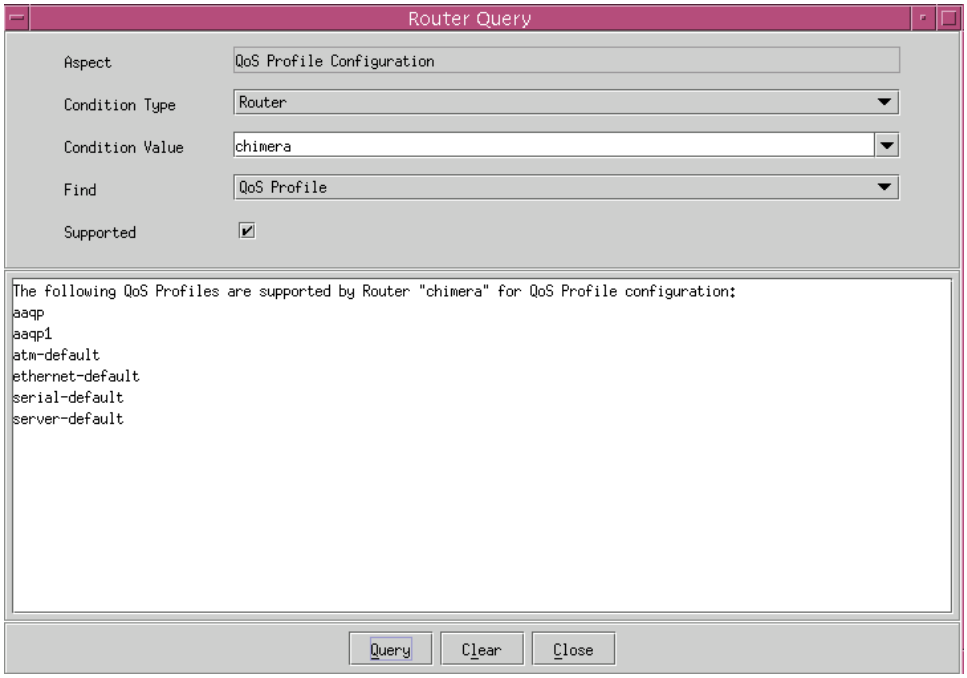
Supported

- Whether or not to search for the condition type that exists or does not exist on the router, QoS profile, or policy group.
- Value—Checked or unchecked
  - Checked—Searches for the condition type that is on the router, QoS profile, or policy group
  - Unchecked—Searches for the condition type that is not on the router, QoS profile, or policy group
- Default—No value

Examples: Searching for QoS Information

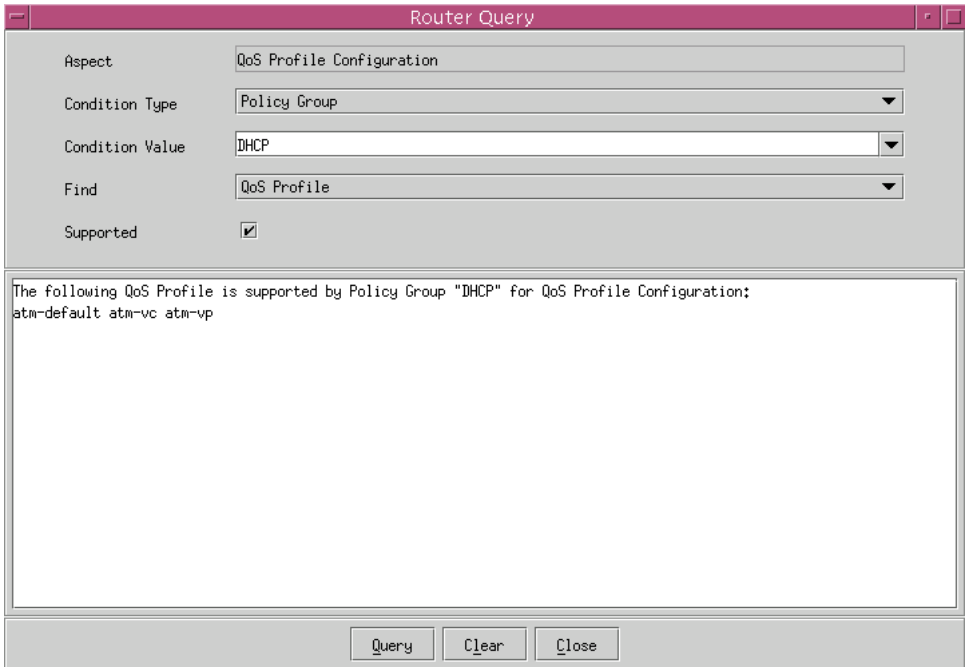
The query example in Figure 1 on page 14 searches for all QoS profiles on router chimera.

Figure 1: Searching for All QoS Profiles on a Router



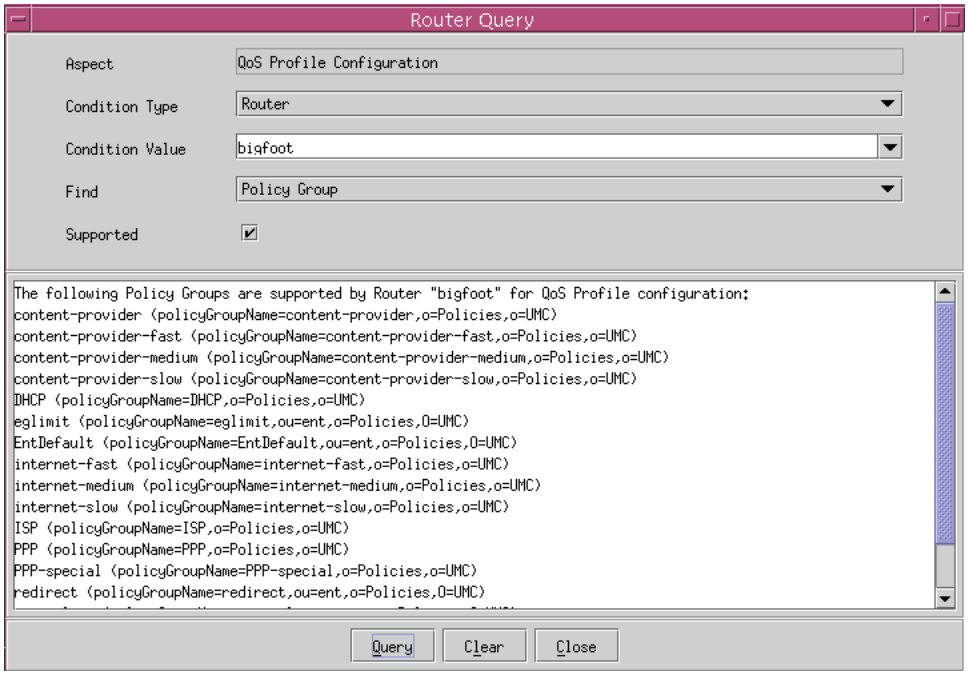
The query in Figure 2 on page 14 searches for QoS profiles in policy group DHCP.

Figure 2: Searching for QoS Profiles in a Policy Group



The query in Figure 3 on page 15 searches for all policy groups that router bigfoot supports. For a policy group to be supported on a router, both the policy group and the router must contain the same QoS profile.

Figure 3: Searching for All Policy Groups on a Router



Related Topics

- Dynamically Managing QoS Profiles on page 4
- Policy Management Overview
- Policy Components
- Overview of QoS on JUNOSe Routers on page 3

CHAPTER 2

Managing Subscribers for a Wireless Roaming Environment

Overview of a Wireless Roaming Environment

In a roaming wireless environment, subscribers can log in to a wireless access point at a variety of wireless locations owned by service providers that participate in a roaming network agreement. The wireless locations participating in the agreement can be owned by one or more service providers.

Typically, RADIUS manages information about subscribers between the wireless locations. A RADIUS server for an Internet service provider (ISP) manages authentication for its subscribers, and shares information with the other ISPs with which the service provider has a roaming agreement. Subscribers can log in to an SAE from any supported site.

The SAE provides support for RADIUS vendor-specific attributes for wireless Internet service provider roaming (WISPr).

Related Topics

- Subscriber Access in a Wireless Roaming Environment on page 17
- Configuring Subscriber Access for a Wireless Location on page 18
- For more information RADIUS vendor-specific attributes for wireless Internet service provider roaming (WISPr): http://www.wi-fialliance.org/opensection/wispr.asp

Subscriber Access in a Wireless Roaming Environment

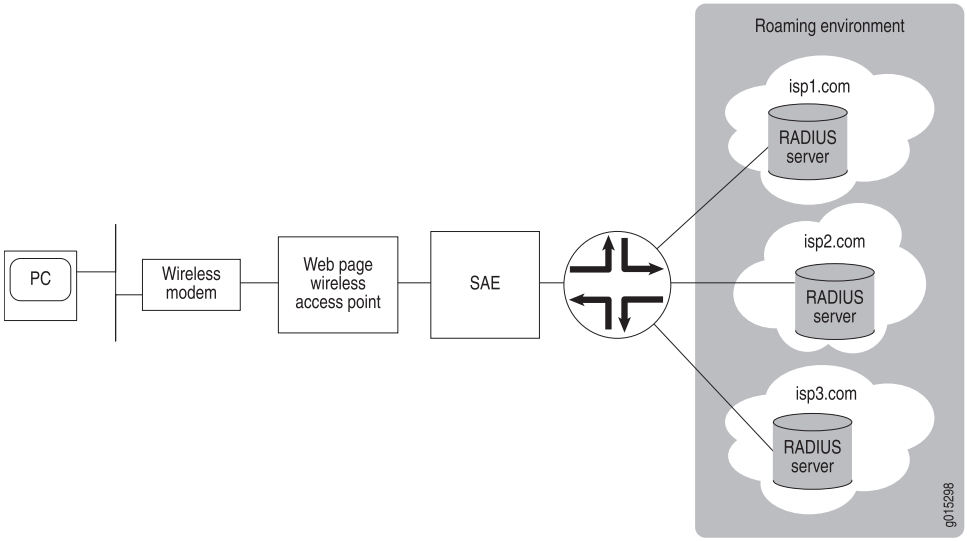
When subscribers log in to a wireless location that has a roaming agreement with other locations, the following sequence of events occurs:

1. Subscribers connect to the local wireless location and provide login information on a portal page that provides a universal access method. This login information is forwarded to the SAE.
2. Based on the login information, an access service starts.

3. The subscriber is authenticated by RADIUS; the authorization includes RADIUS vendor-specific attributes for WISPr.
4. Policies are activated for the subscriber on the router.
5. After successful start of the access service, the portal page redirects the subscriber to a specified start page.

Figure 4 on page 18 shows how subscribers interact with an SAE-managed wireless location that has a roaming agreement with wireless locations.

Figure 4: Subscriber Access to a Wireless Roaming Group



Related Topics

- Overview of a Wireless Roaming Environment on page 17
- Configuring Subscriber Access for a Wireless Location on page 18

Configuring Subscriber Access for a Wireless Location

Tasks to use the SAE to manage a wireless access point that participates in a roaming agreement are:

1. Configuring RADIUS Authentication on page 18
2. Creating Subscriber Access to an ISP on page 21
3. Creating Web Access on page 22
4. Setting Idle Timeout Options for the SAE on page 23

Configuring RADIUS Authentication

You configure RADIUS authentication for users who connect from a wireless location, and set up RADIUS authentication to support a roaming environment between wireless Internet service providers. You can use the Flexible RADIUS Authentication plug-in that is provided with the SRC software, or you can create a custom RADIUS authentication plug-in.

Configuring a Custom RADIUS Authentication Plug-In

If you create a custom plug-in, be sure that it supports the same RADIUS attributes as those configured for the flexible RADIUS authentication plug-in. See "Configuring the Flexible RADIUS Authentication Plug-In" on page 19 .

For information about creating a custom plug-in, see *SAE CORBA Plug-In Service Provider Interface (SPI)* on the Juniper Networks Web site at:
http://www.juniper.net/techpubs/software/management/src/api-index.html

Configuring the Flexible RADIUS Authentication Plug-In

The default flexible RADIUS authentication plug-in, flexRadiusAuth, provides support for RADIUS vendor-specific attributes for WISPr, which are listed in the following procedure. These attributes use the IANA private enterprise number 14122 assigned to the Wi-FI Alliance. For more information about these attributes, see http://www.wi-fialliance.org/opensection/wispr.asp

You should be familiar with the general procedure for configuring the flexible RADIUS authentication plug-in before configuring it to include the WISPr attributes. For information about configuring the flexible RADIUS authentication plug-in, see Configuring Tracking Plug-Ins (SRC CLI).

When you configure the plug-in, you can use the following standard attribute values to set values in authentication response packets:

- setAcctInterimTime
- SetSubstitution
- SetTerminateTime

Examples in the following procedure show how you can use these attribute values.

To configure the plug-in to support a roaming environment:

1. Configure attributes.
   - Required attributes:
     - An identifier for the wireless location:

       vendor-specific.WISPr.Location-ID=*Identifier*

       This attribute can be an interface description (ifAlias) or other value that identifies the JUNOSe interface to which the wireless access point connects.
     - The URL of the start page returned by the RADIUS server of the ISP:

       vendor-specific.WISPr.Redirection-URL=*Command to make the URL available to the SRC software*

       For example:

       vendor-specific.WISPr.Redirection-URL=setProperty(" startURL=%s" % ATTR)

       The default configuration sets a session property named startURL.
     - The URL of a page that a subscriber can use to log out of the network:

       vendor-specific.WISPr.Logoff-URL=*URL of a log out page*

- Bandwidth attributes (recommended):
  - The maximum transmission rate in bits per second:

    vendor-specific.WISPr.Bandwidth-Max-Up=*Command to make the rate available to the SRC software*

    For example:

    vendor-specific.WISPr.Bandwidth-Max-Up=setSubstitution(“ max_up_rate=%s” % ATTR)
  - The maximum receive rate in bits per second:

    vendor-specific.WISPr.Bandwidth-Max-Down=*Command to make the rate available to the SRC software*

    For example:

    vendor-specific.WISPr.Bandwidth-Max-Down=setSubstitution(“ max_down_rate=%s” % \ ATTR)
- Optional attributes:
  - The name of the wireless location:

    vendor-specific.WISPr.Location-Name=*Name of the wireless location*
  - The date and time that the subscriber session is to end:

    vendor-specific.WISPr.Session-Terminate-Time=*Command to set the session terminate time*

    For example:

    vendor-specific.WISPr.Session-Terminate-Time=setTerminateTime(ATTR)
  - The end of the subscriber session at the end of the billing day:

    vendor-specific.WISPr.Session-Terminate-End-Of-Day=ATTR or setTerminateTime("00:00:00")

    If the operator of the wireless location does not support daily billing, do not configure this attribute, and remove it if present.
  - A service type for billing:

    vendor-specific.WISPr.Billing-Class-Of-Service=*Service type*

2. For each attribute that you configure, configure the packet type to which the attribute applies. Table 5 on page 20 shows the packet types associated with each attribute.

Table 5: Packet Types for RADIUS Attributes

RADIUS Attribute	Associated RADIUS Packet Definition
vendor-specific.WISPr.Location-ID	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Location-ID
vendor-specific.WISPr.Redirection-URL	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Redirection-URL
vendor-specific.WISPr.Logoff-URL	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Logoff-URL

Table 5: Packet Types for RADIUS Attributes *(continued)*

RADIUS Attribute	Associated RADIUS Packet Definition
vendor-specific.WISPr.Bandwidth-Max-Up	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Bandwidth-Max-Up
vendor-specific.WISPr.Maximum-Max-Down	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Maximum-Max-Down
vendor-specific.WISPr.Location-Name	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Location-Name
vendor-specific.WISPr.Session-Terminate-Time	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Session-Terminate-Time
vendor-specific.WISPr.Session-Terminate-End-Of-Day	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Session-Terminate-End-Of-Day
vendor-specific.WISPr.Billing-Class-Of-Service	RadiusPacket.stdAuth.auth.vendor-specific.WISPr.Billing-Class-Of-Service

Creating Subscriber Access to an ISP

Configure a service that lets subscribers connect to an ISP through a captive portal, a single Web page to which subscribers connect. The policies associated with the service should specify a JUNOS policing or JUNOSe rate-limiting policy to set the maximum bandwidth at which:

- A subscriber can send traffic.
- A subscriber can receive traffic.

When you configure the policies, define the bandwidth values as parameters so that the policies can be applied across a number of subscribers.

To configure a service to access the ISP:

1. Create the SRC service to use RADIUS authentication.

   See Adding a Normal Service (SRC CLI).

2. Create a policy group the sets the maximum bandwidth at which a subscriber can send traffic, and the maximum bandwidth at which a subscriber can receive traffic. Use parameters to set these values.

   To configure policies , see:

   - Configuring Policy Groups (SRC CLI) on page 193
   - Configuring Global Parameters (SRC CLI)
   - Configuring Local Parameters (SRC CLI)

For example, you can create a policy configuration that includes:

- A local parameter named max_up_rate that sets the maximum rate at which the subscriber can send data
- A local parameter named max_down_rate that sets the maximum rate at which the subscriber can receive data
- A policy group Receive(Downstream) that references max_down_rate
- A policy group Send(Upstream) that references max_up_rate

Substitutions for these parameters can then be referenced in the RADIUS attributes:

```
vendor-specific.WISPr.Bandwidth-Max-Up=setSubstitution(“ max_up_rate=%s” % ATTR)
vendor-specific.WISPr.Bandwidth-Max-Down=setSubstitution(“ max_down_rate=%s”
  % ATTR)
```

Creating Web Access

When subscribers connect to and log in to a wireless access point, they are directed to a single Web page that is referred to as a captive portal page. This page is part of a service selection portal. A captive portal page receives and manages redirected Web requests. The SRC Application Library provides an unsupported, demonstration application for a residential service selection portal.

When creating a captive portal page for a wireless roaming environment, configure the page to:

- Start an access service that is configured to be authenticated by the RADIUS server of the ISP.
- After the access service starts, redirect the subscriber to the page specified by the Redirect-URL RADIUS attribute. This page is the start page for the subscriber's home ISP.

  You can retrieve the URL of the start page from the service session property startURL. Note that startURL is the default name used for the flexible RADIUS authentication plug-in; you can assign a different name to this property.

  You can use the Subscriber.readSubscription() method in the Common Object Request Broker Architecture (CORBA) remote application programming interface (API) to retrieve the redirect URL.

Note that when you develop the portal, you can use the following methods in the SAE CORBA remote API to retrieve session data after the access service starts:

- Subscriber.readSubscriber()
- Subscriber.readSubscription()

For more information about these methods, see the SAE CORBA remote API documentation on the Juniper Networks Web site at

http://www.juniper.net/techpubs/software/management/src/api-index.html.

Setting Idle Timeout Options for the SAE

You can configure the following options to ensure that the timeout values are consistent with the requirements for your environment:

- Idle timeout—Defines how long a session is idle before the connection is closed.
- Adjust session time—Adjusts the session time reported in an accounting messaged by subtracting idle time from the time if the session times out.

To configure the timeout settings:

1. Configure the service activation authentication through a RADIUS server to return an idle timeout. This configuration requires that the RADIUS server returns the idle timeout vendor-specific attribute (VSA).

   or

   Configure the idle timeout in the SRC service definition. For example:

   ```
   [edit services global service service1]
   user@host# set idle-timeout 5
   ```

   Although an interval up to 5 minutes is typically recommended, for the SRC software, we recommend a minimum of 15 minutes.

2. Configure the **adjust-session-time statement** for the SAE to ensure that session time is accurately reported for accounting purposes. For example:

   ```
   [edit shared sae group wireless configuration]
   user@host# set idle-timeout adjust-session-time
   ```

Related Topics

- Overview of a Wireless Roaming Environment on page 17
- Subscriber Access in a Wireless Roaming Environment on page 17
- *JUNOS System Basics Configuration Guide*

http://www.juniper.net/techpubs/software/junos/junos84/swconfig84-system-basics/swconfig84-system-basics.pdf

- *JUNOSe Broadband Access Configuration Guide*

http://www.juniper.net/techpubs/software/erx/junose82/bookpdfs/swconfig-broadband.pdf

CHAPTER 3

Configuring VoIP Services in an SRC Network

Overview of Session Management for VoIP Services

When the SAE activates a service session, it authorizes the session with authorization plug-ins; it may use the admission control plug-in (ACP) to perform call admission control and allocate bandwidth; and it installs the policy required for the service on a JUNOSe interface.

VoIP and multimedia service sessions are typically established in multiple phases that require changes to installed policies and authorized bandwidth while the service session remains active. To support VoIP sessions, the SAE allows changes to active service sessions. These changes include:

- Controlled bandwidth. If bandwidth demand increases, the authorization plug-in must authorize the change.
- Policy parameters. Only parameter substitution values can be changed. Policy parameters can include classifiers, such as destination address and port, and actions, such as rate-limit profiles.
- Session and idle timeouts. All attributes that can be set for initial service activation can be set for service session modifications.

Accounting and Tracking

Accounting information is preserved across service session changes. Accounting information for a complete service session includes the sum of counters for all service session segments.

When the ACP receives an interim update request, it compares the upstream and downstream bandwidth in the request with the current values. If the bandwidth has

changed, ACP modifies its counters based on the difference between the current and new values.

Tracking plug-ins are informed of service session changes through an interim update message. The interim update is sent even if regular interim updates are disabled. If the controlled bandwidth changes, the interim update message contains the new bandwidth settings.

VoIP Call Setup

Initial setup of a VoIP call requires changes to bandwidth and to the endpoint address during call setup. The setup sequence for a VoIP call can follow this pattern:

1. The subscriber attempts to establish a call.
2. The gatekeeper (or Session Initiation Protocol [SIP] proxy) performs local admission control.
3. The gatekeeper allocates a Codec for the call; for example, 64 kbps.
4. The gatekeeper activates the VoIP service on the SAE with 64 kbps bandwidth and a destination address of unknown.
5. The SAE performs admission control, activates a service session, and installs policies on the router.
6. The gatekeeper negotiates call parameters with the remote endpoint.
7. The gatekeeper modifies the VoIP service with negotiated parameters; for example, 32 kbps, destination address 10.10.3.4, and UDP port 5678.
8. The SAE creates new policies that reflect changes to the traffic classifier and rate-limit profile, and then removes the existing policies from the router and installs the new policies.
9. The SAE sends interim updates to the ACP and tracking plug-ins.

Related Topics

- Overview of Global and Local Parameters
- For information about configuring and managing policies, see the *SRC PE Services and Policies Guide*
- Configuring Policies and Services for VoIP on page 26
- Activating VoIP Services for Assigned IP Subscribers on page 27

Configuring Policies and Services for VoIP

When you set up a service that supports VoIP, you need to create a policy group for the VoIP service and assign the policy group to the VoIP service.

The SAE installs the policy on the router when the service is activated. When the service session is modified during VoIP call setup, the SAE replaces policy values with new values that were negotiated during call setup. The SAE then creates a new policy and installs it on the router.

When you set up a policy group for VoIP services, you need to assign variable parameters to fields that the SAE will need to modify. For example, source and destination addresses and UDP ports might be replaced with actual values. Upstream and downstream rate-limit parameters, such as committed rate and burst sizes, are likely to be modified.

Related Topics

- Overview of Session Management for VoIP Services on page 25
- Configuring Policy Groups (SRC CLI) on page 193
- Activating VoIP Services for Assigned IP Subscribers on page 27

Activating VoIP Services for Assigned IP Subscribers

When the SAE activates VoIP services, signaling proxies must identify subscriber equipment based on the IP address of the equipment. In the enterprise model, an IT manager typically subscribes to a service at a particular level in the subscriber hierarchy, and then provides the service to all access lines and subscribers who are at lower levels in the hierarchy. In cases such as this, the SAE manages the router interface but not the subscriber. The SAE does not know the IP addresses of the subscribers and therefore cannot provide the IP address to the signaling proxies.

A type of subscriber session called assigned IP supports the case in which the SAE does not manage the subscriber but needs to provide the IP address to signaling proxies. The SAE dynamically creates an assigned IP session based on an API call. The VoIP gateway must provide the following information to the SAE before the SAE can create the assigned IP session:

- The subscriber's IP address
- The name of a managed interface (The SAE applies policies for service sessions to this interface.)
- The name of the virtual router in which the managed interface resides

The NIC maps the subscriber's IP address to the SAE reference of the managing SAE, the interface name, and the virtual router name and provides this information to the VoIP gateway.

The network information collector (NIC) keeps track of managed interfaces through a NIC SAE plug-in agent. When an interface start, stop, or interim update event occurs, the SAE sends the interface tracking events to the NIC SAE plug-in agent. The NIC uses this information as part of the process of creating these mappings.

Related Topics

- Overview of Session Management for VoIP Services on page 25
- Configuring the NIC (SRC CLI)
- Configuring Policies and Services for VoIP on page 26
- Setting Timeouts for Assigned IP Subscriber Sessions on page 28

Setting Timeouts for Assigned IP Subscriber Sessions

To set timeouts for assigned IP subscriber sessions in the SAE configuration:

1. From configuration mode, access the SAE configuration statement that configures subscriber sessions.

   ```
   [edit]
   user@host# edit shared sae configuration subscriber-sessions
   ```

2. Specify the interval after which assigned IP subscriber sessions are deactivated if no service session is active.

   ```
   [edit shared sae configuration subscriber-sessions]
   user@host# set assigned-ip-idle-timeout assigned-ip-idle-timeout
   ```

Related Topics

- Overview of Session Management for VoIP Services on page 25
- Tracking and Controlling Subscriber and Service Sessions with SAE APIs
- Configuring Access to Subscriber Data (SRC CLI)
- Activating VoIP Services for Assigned IP Subscribers on page 27

CHAPTER 4

Providing Packet Mirroring in an SRC Network

Overview of Packet-Mirroring Services

Packet mirroring allows you to mirror subscriber traffic by configuring a script service with the SRC software that applies policies on a router running JUNOSe Software for RADIUS-based packet mirroring.

When the SAE activates a packet-mirroring service session, the session sends dynamic RADIUS requests, such as change-of-authorization (CoA) messages, to a RADIUS device such as a router running JUNOSe Software.

In RADIUS-based packet mirroring on a router running JUNOSe Software, a RADIUS administrator uses RADIUS attributes to configure packet mirroring of a particular subscriber's traffic. The router creates dynamic secure policies for the mirroring operation. The original traffic is sent to its intended destination, and the mirrored traffic is sent to an analyzer device (the mediation device). The mirroring operations are transparent to the subscriber whose traffic is being mirrored. This dynamic method uses RADIUS attributes and RADIUS vendor-specific attributes (VSAs) to identify a subscriber whose traffic is to be mirrored and to trigger the mirroring session. RADIUS-based mirroring uses dynamically created secure policies based on certain RADIUS VSAs. You attach the secure policies to the interface used by the mirrored subscriber. The packet-mirroring VSAs that the RADIUS server sends to the E Series router are MD5 salt-encrypted.

You must deploy RADIUS-based packet mirroring on routers running JUNOSe Software to monitor the subscriber traffic.

Related Topics

- Configuring Packet-Mirroring Support in an SRC Network on page 30
- Configuring the Script Service for Packet Mirroring on page 31
- Configuring Parameters for the Script Service for Packet Mirroring on page 32
- Example: Using the Sample Packet-Mirroring Application on page 35

Configuring Packet-Mirroring Support in an SRC Network

To support packet mirroring in an SRC network, configure a script service that can be activated to set up RADIUS-based packet-mirroring policies on a router running JUNOSe Software. The script service defines the parameters needed to mirror subscriber traffic, such as the address of the subscriber or the analyzer device. This script service is activated for the subscriber whose traffic should be mirrored.

You must have preconfigured RADIUS-based packet mirroring on routers running JUNOSe Software. The JUNOSe software provides RADIUS-based packet mirroring, which allows the router to create dynamic secure policies for the mirroring operation. The RADIUS administrator can configure and manage interface mirroring services that are activated by means of CoA.

To set up the SRC software for packet mirroring:

- Create a script service for packet mirroring.

  The SRC software includes a sample script service that you can configure to send dynamic RADIUS requests to the router running JUNOSe Software. You can use the sample service definition and customize it for your environment by modifying the service substitutions.

  See “Configuring Parameters for the Script Service for Packet Mirroring” on page 32.

- Configure subscriptions to the packet-mirroring service.

  You can set up the subscriptions to activate immediately on login.

  See Configuring Subscriptions (SRC CLI).

- (Optional) Configure the maximum number of RADIUS peers.

  See “Specifying Maximum Number of RADIUS Peers (SRC CLI)” on page 34.

Related Topics

- For information about configuring RADIUS-based packet mirroring on the router running JUNOSe Software, see the *JunosE Policy Management Configuration Guide*
- For information about dynamic RADIUS requests, see RFC 3576—Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS) (July 2003)
- Configuring the Script Service for Packet Mirroring on page 31
- Example: Using the Sample Packet-Mirroring Application on page 35
- Overview of Packet-Mirroring Services on page 29

Configuring the Script Service for Packet Mirroring

To configure the script service for packet mirroring:

1. Create a script service in the services global service ***name*** hierarchy or the services scope ***name*** service ***name*** hierarchy. For example:

   ```
   [edit]
   user@host# edit services global service  packetMirroring
   ```

2. Set the type to script.

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

3. (Optional) Configure other properties as needed for your service.
4. Configure the script properties.
   a. Access the script hierarchy for the configured script service.

      ```
      [edit services global service packetMirroring]
      user@host# edit script
      ```

   b. Specify URL as the script type.

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

   c. Specify the name of the Java class that implements the script service.

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

   d. Configure the URL of the script service or the path and filename of the service.

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

5. Verify the configuration.

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

6. Configure the parameters for the script service.

   See “Configuring Parameters for the Script Service for Packet Mirroring” on page 32.

Related Topics

- Configuring Packet-Mirroring Support in an SRC Network on page 30

- Adding a Normal Service (SRC CLI)
- Customizing Service Implementations
- Example: Using the Sample Packet-Mirroring Application on page 35
- Overview of SRC Script Services
- Overview of Packet-Mirroring Services on page 29

Configuring Parameters for the Script Service for Packet Mirroring

Provide parameter substitutions with the values that are in the service definitions for the script service.

Table 6 on page 32 lists the parameters specified by the sample packet-mirroring script service. In most cases, you can use the sample script service without modification.

Table 6: Parameter Substitutions for Packet-Mirroring Services

Parameter Name	Description
dynAnalyzerIPAddress	RADIUS VSA that is the IP address of the analyzer device. This attribute is required.
dynAnalyzerPortNumber	RADIUS VSA that is the UDP port number of the monitoring application in the analyzer device. If specified, dynMirrorIdentifier must also be specified.
dynMirrorIdentifier	RADIUS VSA in the form of a hexadecimal string. If specified, dynAnalyzerPortNumber must also be specified.
dynClientIp	IP address of the dynamic RADIUS client.
dynClientPort	UDP port number of the dynamic RADIUS client.
dynServerIp	IP address of the C Series Controller.
dynServerPort	UDP port number of the C Series Controller.
dynSecret	Shared secret.
dynRetry	Number of retries for sending dynamic RADIUS packet when no RADIUS response is received. The retry interval is 3 seconds.

Table 6: Parameter Substitutions for Packet-Mirroring Services *(continued)*

Parameter Name	Description

dynConfig	Content of dynamic RADIUS request packets 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.
	  • JUNOSe VSA in one of the following formats:
	  vendor-specific.4874.<vsa#>[.salt]
	  26.4874.<vsa#>[.salt]
	  where .salt indicates that the attribute is MD5 salt-encrypted in the RADIUS packet.
	• pluginEventAttribute—Valid Python expression
	• \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.4874.58.salt=1\nstart.vendor-specific.JUNIPER.Unisphere-Med-Dev-Handle.salt=custom['dynMirrorIdentifier']\nstart.vendor-specific.JUNIPER.Unisphere-Med-Ip-Address.salt=intIp(custom['dynAnalyzerIPAddress'])\nstart.vendor-specific.JUNIPER.Unisphere-Med-Port-Number.salt=int(custom['dynAnalyzerPortNumber'])\nstop.vendor-specific.4874.58.salt=0"

To configure substitutions for the script parameters:

1. At the hierarchy for the script service, specify substitutions for the parameters. For example:

```
[edit services global service packetMirroring]
user@host# set parameter substitution [ dynAnalyzerIPAddress=10.227.6.221
  dynAnalyzerPortNumber=9100 dynMirrorIdentifier=0x0000000100000001
  dynSecret=\"secret\" dynRetry=2 dynClientIp=10.227.7.111 dynClientPort=9099
  "dynConfig=\"start-stop.Acct-Session-Id =
  ifSessionId\\nstart.vendor-specific.JUNIPER.Unisphere-LI-Action.salt=1\\nstar
  t.vendor-specific.JUNIPER.Unisphere-Med-Dev-Handle.salt=custom['dynMirrorIde
  ntifier']\\nstart.vendor-specific.JUNIPER.Unisphere-Med-Ip-Address.salt=intIp(c
  ustom['dynAnalyzerIPAddress'])\\nstart.vendor-specific.JUNIPER.Unisphere-Me
  d-Port-Number.salt =
  int(custom['dynAnalyzerPortNumber'])\\nstop.vendor-specific.JUNIPER.Unisph
  ere-LI-Action.salt=0\"" ]
```

2. Verify the configuration.

```
[edit services global service packetMirroring]
user@host# show
```

```
type script;
status active;
parameter {
  substitution [ dynAnalyzerIPAddress=10.227.6.221
dynAnalyzerPortNumber=9100 dynMirrorIdentifier=0x0000000100000001
dynSecret=secret dynRetry=2 dynClientIp=10.227.7.111 dynClientPort=9099
"dynConfig=\"start-stop.Acct-Session-Id =
ifSessionId\\nstart.vendor-specific.JUNIPER.Unisphere-LI-Action.salt=
1\\nstart.vendor-specific.JUNIPER.Unisphere-Med-Dev-Handle.salt=
custom['dynMirrorIdentifier']\\nstart.vendor-specific.JUNIPER.Unisphere-Med-Ip-Address.salt=

intIp(custom['dynAnalyzerIPAddress'])\\nstart.vendor-specific.JUNIPER.Unisphere-Med-Port-Number.salt
 =
int(custom['dynAnalyzerPortNumber'])\\nstop.vendor-specific.JUNIPER.Unisphere-LI-Action.salt=0\""
 ];
}
script {
  script-type url;
  class-name net.juniper.smgt.scriptServices.packetMirroring.LiService;
  file file:///opt/UMC/sae/lib/pm.jar;
}
```

Related Topics

- Configuring Packet-Mirroring Support in an SRC Network on page 30
- Adding a Normal Service (SRC CLI)
- Setting Parameter Values for Services (SRC CLI)
- Customizing Service Implementations
- Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API on page 36

Specifying Maximum Number of RADIUS Peers (SRC CLI)

The dynamic RADIUS server can maintain a certain number of peers.

To specify the maximum number of peers with the SRC CLI:

1. From configuration mode, access the SAE configuration statement that configures dynamic RADIUS options.

   ```
   [edit]
   user@host# edit shared sae configuration dynamic-radius-server
   ```

2. Specify the maximum number of peers maintained by the dynamic RADIUS server.

   ```
   [edit shared sae configuration dynamic-radius-server]
   user@host# set maximum-cached-peer maximum-cached-peer
   ```

Related Topics

- Configuring Packet-Mirroring Support in an SRC Network on page 30
- Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API on page 36
- Example: Using the Sample Packet-Mirroring Application on page 35

- Overview of Packet-Mirroring Services on page 29

Example: Using the Sample Packet-Mirroring Application

To use the sample packet-mirroring application:

1. Download the SRC sample applications to your system from the Juniper Networks Web site:

   http://www.juniper.net/support/csc/swdist-erx/src.html

2. Locate the file that contains the service definition:

   /SDK/scriptServices/packetMirroring/ldif/service.ldif

3. Import the sample service definition to the Juniper Networks Database on the C Series Controller. To load the sample data into the database, you can use an LDAP tool, such as **ldapadd**.

   You can obtain **ldapadd** from the following Web site:

   http://www.openldap.org/

   To load data into the Juniper Networks database, you need the IP address of the database and the database credentials. The default bind distinguished name (DN) for the database is *cn=umcadmin, o=umc* and the password is admin123.

4. Copy the */lib/pm.jar* file used by the script service to the */opt/UMC/sae /var/run* directory on the C Series Controller.

5. Modify the service substitutions for your environment.

   You can make these substitutions by defining the parameter substitutions in the packetMirroring service (*serviceName=packetMirroring, o=Services, o=umc*) with the SRC CLI or by passing the values through the SAE core API.

   For information about parameter substitutions, see “Configuring Parameters for the Script Service for Packet Mirroring” on page 32. For information about passing the values through the SAE core API, see “Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API” on page 36.

6. Configure a subscription to the packetMirroring service that is activated on login.

   For information about subscriptions, see Overview of Subscriptions.

7. If you are modifying the sample application, copy the *sae.jar* and *logger.jar* files from the *SKD/lib* directory, and add the *sae.jar* and *logger.jar* files to the classpath when you compile your application.

Example: Packet Mirroring for PPP Subscribers

When a PPP subscriber is subscribed to the packet-mirroring service, configure the service as an activate-on-login service at user connection time. After the subscriber has logged in through the SAE remote API, the packet-mirroring service can be subscribed to the PPP subscriber and activated. When the service is activated, a CoA request is sent to the

router running JUNOSe Software that includes the PPP subscriber's accounting session ID to start packet mirroring for this subscriber.

Example: Packet Mirroring for DHCP Subscribers

When a DHCP subscriber is subscribed to the packet-mirroring service, configure the service as an activate-on-login service at user connection time. After the subscriber has logged in through the SAE remote API, the packet-mirroring service can be subscribed to the DHCP subscriber and activated. When the service is activated, a CoA request is sent to the router running JUNOSe Software that includes the DHCP subscriber's IP address and virtual router name for the router running JUNOSe Software to start packet mirroring for this subscriber.

Configuring DHCP Subscriber Sessions

You can use DHCP option 82 to identify the subscriber session. For example, if you set DHCP option 82 as the user login name, an external application can use this setting to search for the subscriber session. The following subscriber classification script illustrates this example:

```
[radname=default=Users=UMC?loginName=<-dhcp[82]suboptions[1]string->?sub?(interfaceName=<-dhcp[82]suboptions[1]string->)]
loginType = " ADDR"
[<-retailerDN->??sub?(uniqueID=<-userName->)]
retailerDN != " "
& userName != " "
[<-unauthenticatedUserDn->]
loginType == "ADDR"
loginType == "AUTHADDR"
```

Disabling RADIUS Authentication for DHCP Subscribers

Packet mirroring for DHCP subscribers does not involve RADIUS authentication, so you might have to configure authentication to grant all IP subscriber management interfaces access without authentication. For example, configure the router running JUNOSe Software with the following authentication:

```
aaa authentication ip default none
```

You can still configure other subscribers to use RADIUS authentication. For example, configure the router running JUNOSe Software with the following authentication for PPP subscribers:

```
aaa authentication ppp default radius
```

Related Topics

- Configuring Packet-Mirroring Support in an SRC Network on page 30
- Overview of Packet-Mirroring Services on page 29

Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API

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

- Service definition

- 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 running JUNOSe Software that is attached to the service session.

Related Topics

- Configuring Parameters for the Script Service for Packet Mirroring on page 32
- 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/packetMirroring/java/net/juniper/smgt/scriptServices/packetMirroring/LiService.java

PART 2

Managing Services in a PCMM Environment

CHAPTER 5

Providing Premium Services in a PCMM Environment

Overview of a PCMM Environment

The PCMM specification defines a standards-based way to deliver premium quality of service (QoS)—enhanced services across the radio frequency (RF) portion of a cable network. The PCMM capabilities of the SRC software along with Juniper Networks routers provide an end-to-end solution that seamlessly links the cable operator's RF domain with IP edge and core QoS services.

Key services supported in this environment include:

- Bandwidth on demand and variable bandwidth
- QoS-enabled streaming media, including video on demand and video telephony
- Residential voice over IP (VoIP)
- Multicast audio and video applications
- Videoconferencing
- Interactive gaming
- Peer-to-peer controls and protection services

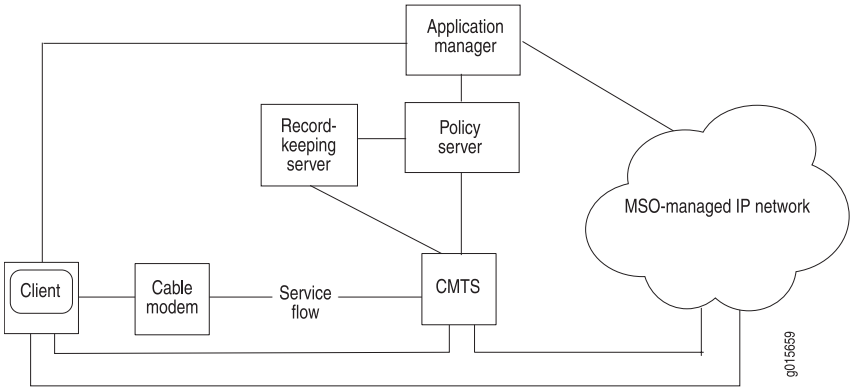
PCMM Architecture

Figure 5 on page 42 depicts the PCMM architectural framework. The basic roles of the various PCMM components are:

- Application manager—Provides an interface to policy server(s) for the purpose of requesting QoS-based service on behalf of a subscriber or a network management system. It maps session requests to resource requests and creates policies.
- Policy server—Acts as a policy decision point and policy enforcement point and manages relationships between application managers and cable modem termination system (CMTS) devices.

- CMTS device—Cable modem termination system. Performs admission control and manages network resources through Data over Cable Service Interface Specifications (DOCSIS) service flows.
- Client—Represents endpoints, such as PC applications, that can send or receive data.
- Record-keeping server—Receives event messages from other network elements, such as the policy server or CMTS device, and acts as a short-term repository for the messages. It can also assemble event messages into coherent sets or call detail records, which are then made available to other back office systems, such as billing, fraud detection, and other systems.

Figure 5: PCMM Architectural Framework



In the PCMM architecture, a client requests a multimedia service from an application manager. The application manager relays the request to a policy server. The policy server is then responsible for provisioning the policies on a CMTS device. Based on the request, the policy server records an event that indicates the policy request. The request can include network resource records, and the policy server can provide the records to a record-keeping server, such as a RADIUS accounting server.

The policy server may also provide functions such as tracking resource usage and tracking the authorization of resources on a per-subscriber, per-service, or aggregate basis.

DOCSIS Protocol

The DOCSIS protocol is the standard for providing quality of service for traffic between the cable modem and CMTS devices. The CMTS device is the headend in the DOCSIS architecture, and it controls the operations of many cable modems. Two channels carry signals between CMTS devices and cable modems:

- Downstream channels—Carry signals from the CMTS headend to cable modems.
- Upstream channels—Carry signals from the cable modems to the CMTS headend.

The DOCSIS protocol defines the physical layer and the Media Access Control (MAC) protocol layer that is used on these channels.

A cable modem usually uses one upstream channel and an associated downstream channel. Upstream channels are shared, and the CMTS device uses the MAC protocol to control the cable modem's access to the upstream channel.

Service Flows

The DOCSIS protocol uses the concept of service flows to support QoS on upstream and downstream channels. A service flow is a unidirectional flow of packets that provides a particular quality of service. Traffic is classified into a service flow, and each service flow has its own set of QoS parameters. The SRC software is compliant with the following upstream service flow scheduling types, as defined in the PacketCable Multimedia Specification PKT-SP-MM-I03-051221.

- Best effort—Used for standard Internet traffic such as Web browsing, e-mail, or instant messaging.
- Non-real-time polling service (NRTPS)—Used for standard Internet traffic that requires high throughput, and traffic that requires variable-sized data packets on a regular basis, such as high-bandwidth File Transfer Protocol (FTP).
- Real-time polling service (RTPS)—Used for applications such as Moving Pictures Experts Group (MPEG) video.
- Unsolicited grant service (UGS)—Used for real-time traffic that generates fixed-size data packets on a periodic basis. Applications include VoIP.
- Unsolicited grant service with activity detection (UGS-AD)—Used for applications such as voice activity detection, also known as silence suppression.

Downstream service flows are defined through a similar set of QoS parameters that are associated with the best-effort scheduling type on upstream service flows.

Client Types

The PCMM specification uses the concept of clients and defines a client as a logical entity that can send or receive data. The SRC software supports type 1 and type 2 clients.

The PCMM specification defines two resource reservation models for each client type—a single phase and a dual phase. The SRC software supports the single-phase model.

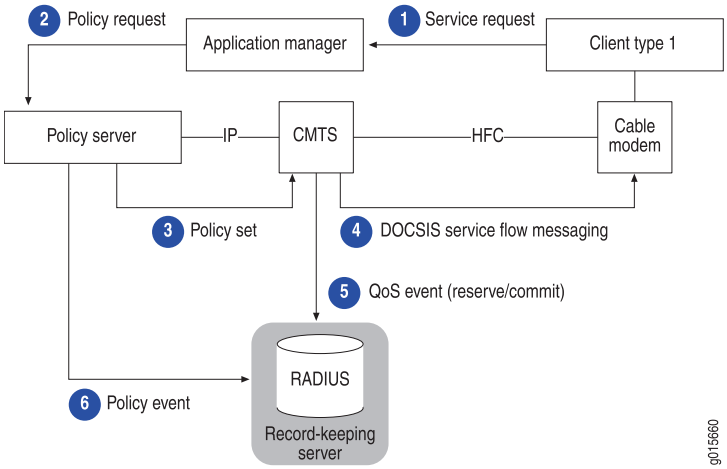
Client Type 1 Single Phase Resource Reservation Model

Type 1 clients represent endpoints, such as PC applications or gaming consoles, that lack specific QoS awareness or signaling capabilities. Type 1 clients communicate with an application manager to request a service. They do not request QoS resources directly from the multiple service operator (MSO) network.

Client type 1 entities support the proxied-QoS with policy-push scenario of service delivery defined in PacketCable Multimedia Architecture Framework Technical Report (PKT-TR-MM-ARCH). In this scenario, the application manager requests QoS resources on behalf of the client, and the policy server pushes the request to the CMTS device. The CMTS device sets up and manages the DOCSIS service flow that the application requires, and might also set up and manage the cable modems.

Figure 6 on page 44 shows the message flow in an application scenario for the client type 1 single-phase resource reservation model.

Figure 6: Client Type 1 Single-Phase Resource Reservation Model



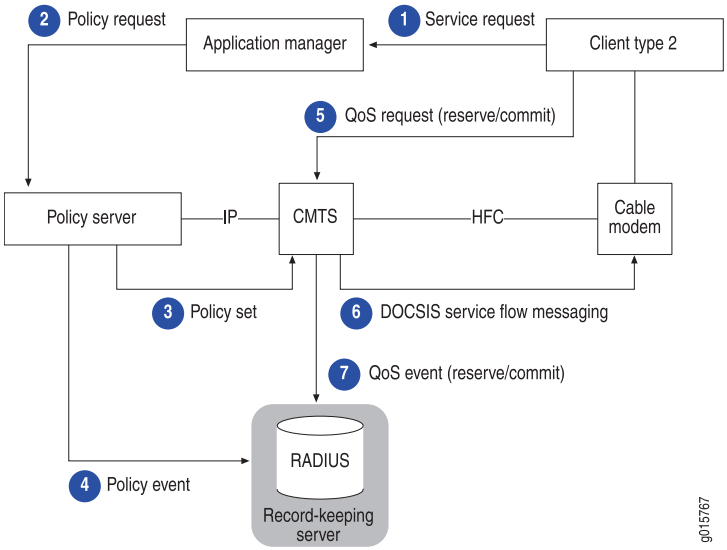
Client Type 2 Single Phase Resource Reservation Model

Type 2 clients represent endpoints that have QoS awareness or signaling capabilities. Type 2 clients communicate with an application manager to request a service and to obtain a token to present for requesting QoS resources directly from the MSO network.

Client type 2 entities support the client-requested QoS with policy-push scenario of service delivery defined in PacketCable Multimedia Architecture Framework Technical Report (PKT-TR-MM-ARCH). In this scenario, the application manager requests QoS resources on behalf of the client, and the policy server pushes the request to the CMTS device. The CMTS device sets up and manages the DOCSIS service flow that the application requires. After the CMTS device sets up the policy, the client can request QoS resources directly from the CMTS device as long as the request is authorized by the policy server.

Figure 7 on page 45 shows the message flow in an application scenario for the client type 2 single-phase resource reservation model.

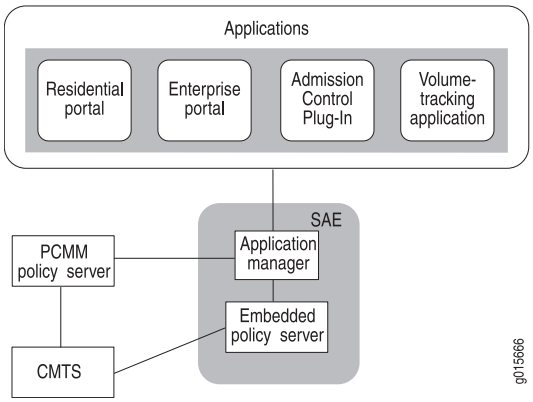
Figure 7: Client Type 2 Single-Phase Resource Reservation Model



SRC Software in the PCMM Environment

Figure 8 on page 45 shows the SRC software in the PCMM environment. The SAE is an application manager that can manage a PCMM-compliant policy server and/or a CMTS device on behalf of applications. The SAE has an embedded policy server that is not fully PCMM-compliant, but it can manage CMTS devices without requiring an external policy server. The Juniper Policy Server (JPS), a component of the SRC software that acts as a policy server, is a PCMM-compliant policy server. For more information about using the JPS, see “JPS Framework” on page 75.

Figure 8: SRC Software in the PCMM Environment



Traffic Profiles

The SRC software supports three types of policies that you can use to define traffic profiles between the CMTS device and the cable modem:

- DOCSIS parameters—Specifies the traffic profile through DOCSIS-specific parameters. You select the type of service flow that you want to offer, and then configure QoS parameters for the service flow.

- Service class name—Specifies the name of a service class that is configured on the CMTS device.
- FlowSpec—Defines the traffic profile through an Resource Reservation Protocol (RSVP)-like parameterization scheme. FlowSpecs support both controlled-load and guaranteed services.

You can also mark packets and then install policies that handle the marked packets in a certain way. The mark action sets the ToS byte in the IP header of IPv4 traffic or the traffic-class field in the IP header of IPv6 traffic.

For more information about traffic profiles, see Delivering QoS Services in a Cable Environment.

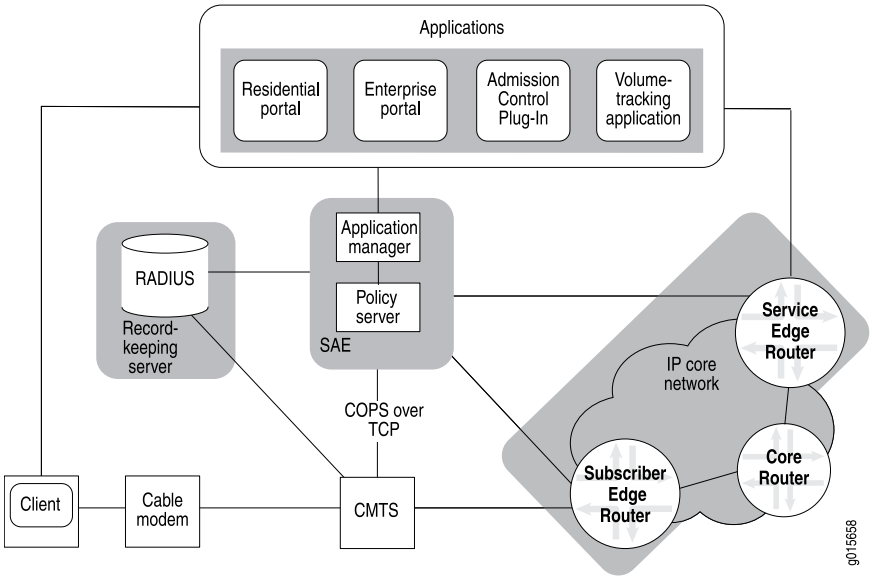
End-to-End QoS Architecture

The previous sections show how the SRC software supports QoS in the cable operator's RF domain, which encompasses the connection from the cable modem to the CMTS device. Using the SRC software along with Juniper Networks routers, you can link the RF domain to the subscriber and service edge domains.

- IP subscriber edge domain—Includes the IP network from the CMTS device to the edge router that typically connects to the cable operator's regional access network. (See "Extending QoS to the Subscriber Edge Domain" on page 47.)
- IP service edge domain—Typically includes the IP network that connects the data center that houses service delivery applications to a backbone or directly to a cable head-on facility. (See "Extending QoS to the Service Edge Domain" on page 47.)

By provisioning services across a network path, you can deliver a particular level of service for specified types of traffic. Figure 9 on page 47 shows a typical high-level architecture of a cable operator and how the SRC software and Juniper Networks routers can be deployed to deliver end-to-end QoS services.

Figure 9: End-to-End QoS Architecture in a Cable Network



Extending QoS to the Subscriber Edge Domain

The subscriber edge domain includes subscriber edge routers that aggregate CMTS devices. To support QoS in subscriber edge domains, QoS must be enabled across the subscriber edge into the core or regional access network. When the SRC software receives a service request, it performs service authorization, which can include admission control. It then sends policies to the appropriate CMTS device and subscriber edge router interface.

In addition to the QoS services required in the RF domain, service policies in the subscriber edge domain that must be available for provisioning at this point include:

- Policy routing to best-of-breed appliances and premium paths
- Rate limiting, traffic shaping, and marking
- Admission control (edge resources and core resources)
- Captive portal and Web redirect capabilities
- Filtering and routers running JUNOS Software-based firewall services
- Routers running JUNOS Software virtual private network (VPN) services

Extending QoS to the Service Edge Domain

The service edge domain includes service edge routers that aggregate applications. To support QoS in service edge domains, the SRC software sends policies to a service edge router that provides for enhanced service delivery to the service origination edge for centralized or hosted services, such as multimedia or VoD.

In addition to the QoS services required in the RF domain, service policies in the service edge domain that must be capable of being provisioned at this point include:

- Policy routing to best-of-breed appliances and premium paths
- Rate limiting, traffic shaping (called hierarchical queuing in JUNOSe software), and marking
- Filtering and routers running JUNOS Software based firewall services
- Routers running JUNOS Software VPN services

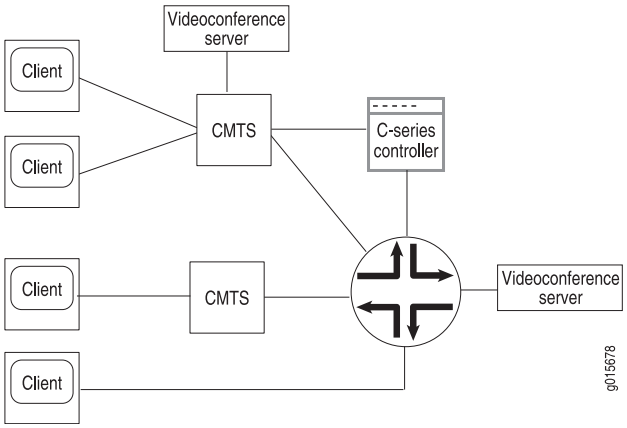
Provisioning End-to-End Services

The following sections provide examples of how you can use the SRC software to provision services for video applications. Although the examples show one SAE managing all the network devices, separate SAEs could manage each device and provide the same service.

Example for Videoconferencing Services

You can configure services to mark traffic forwarded from specified systems, and then apply an end-to-end service level for that traffic. Figure 10 on page 48 shows a scenario in which videoconferencing is delivered in a PCMM environment.

Figure 10: Videoconferencing Example



To ensure a specified level of service from each client PC to the videoconference server and then to each client PC participating in the videoconference, you could configure the following types of services:

- Three services:
  - A service that provides policies to mark packets with a specified type of service for the videoconferencing software.
  - A service that provides policies for the type of service specified for CMTS device.
  - A service that provides policies for the type of service specified for the routers running JUNOS or JUNOSe Software.
- An infrastructure service for each service.
- An aggregate service that contains the three infrastructure services as fragment services.

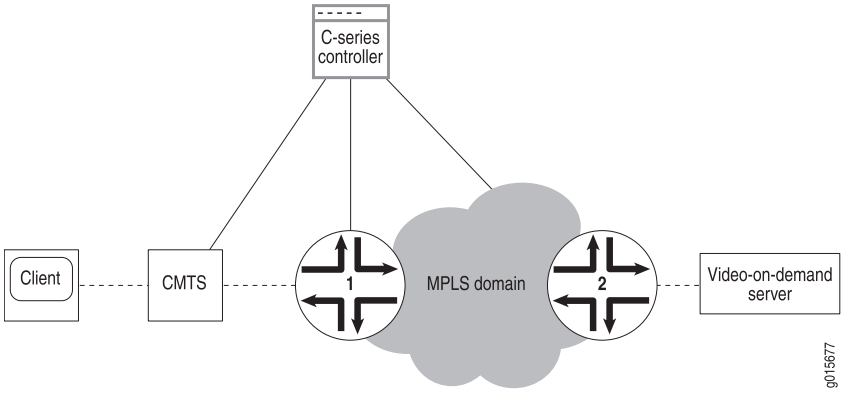
This configuration marks packets that the CMTS device receives from both client and server, and applies forwarding policies on the CMTS device and on the routers running

JUNOSe or JUNOS Software for packets sent to and received from the videoconferencing server.

Example for Video-on-Demand Services

You can configure services to provide server-to-client service for traffic sent from a video-on-demand server to client PCs. Figure 11 on page 49 shows a scenario in which video on demand is delivered in a PCMM environment.

Figure 11: Video-on-Demand Example



To ensure a specified level of service from the video-on-demand server to the client PC, you could configure the following types of services:

- Services that provide bandwidth-on-demand (BoD) policies for traffic that is being forwarded from the video-on-demand server through:
  - Routers running JUNOS Software
  - CMTS devices
- A script service that sets up the Multiprotocol Label Switching (MPLS) path and delivers the specified service level for traffic that is being forwarded from the video-on-demand server through the MPLS domain.
- An infrastructure service for each value-added and script service.
- An aggregate service that contains all the infrastructure services as fragment services.

This configuration applies BoD policies to routers running JUNOSe or JUNOS Software, the MPLS domain, and the CMTS device, and sets up the MPLS path from router running JUNOS Software (2) to router running JUNOS Software (1).

Related Topics

- For more information about each scheduling type, see Delivering QoS Services in a Cable Environment.
- For more information about PCMM, consult the following specifications provided by CableLabs:
  - PacketCable Multimedia Architecture Framework Technical Report (PKT-TR-MM-ARCH)
  - PacketCable Multimedia Specification PKT-SP-MM-I03-051221

Using the SAE in a PCMM Environment

The SAE uses the Common Open Policy Service (COPS) protocol as specified in the PacketCable Multimedia Specification PKT-SP-MM-I03-051221 to manage PCMM-compliant CMTS devices in a cable network environment. The SAE connects to the CMTS device by using a COPS over Transmission Control Protocol (TCP) connection. In cable environments, the SAE manages the connection to the CMTS device.

The CMTS device does not provide address requests or notify the SAE of new subscribers, subscriber IP addresses, or any other attributes. IP address detection and all other subscriber attributes are collected outside of the COPS connection to the CMTS device. The SAE uses COPS only to push policies to the CMTS device and to learn about the CMTS status and usage data.

Because the CMTS device does not have the concept of interfaces, the SRC software uses pseudointerfaces to model CMTS subscriber connections similar to subscriber connections for routers running JUNOS Software.

This section describes how the SAE is used in cable networks. It includes the following topics:

Logging In Subscribers and Creating Sessions

You can use two mechanisms to obtain subscriber address requests and other information and to set up a pseudointerface on the CMTS device. (You must choose one mechanism; you cannot mix them.):

1. Assigned IP subscriber. The SAE learns about a subscriber through subscriber-initiated activities, such as activating a service through the portal or through the Advanced Services Gateway (ASG).

   With this method, you use the assigned IP subscriber login type along with the network interface collector (NIC) to map IP addresses to the SAE.

2. Event notification from an IP address manager. The SAE learns about subscribers through notifications from an external IP address manager, such as a DHCP server or a RADIUS server.

With this method, you use the event notification application programming interface (API). The API provides an interface to the IP address manager, and lets the IP address manager notify the SAE of events such as IP address assignments.

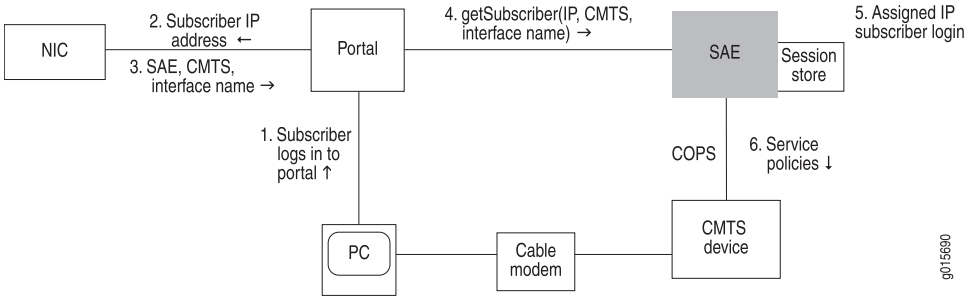
Assigned IP Subscribers

With the assigned IP subscriber method of logging in subscribers and creating sessions, the SRC software uses IP address pools along with NIC resolvers to provide mapping of IP addresses to SAEs. You configure the static address pools or dynamically discovered address pools in the virtual router configuration for a CMTS device. These pools are published in the NIC. The NIC maps subscriber IP addresses in requests received through the portal or Advanced Services Gateway to the SAE that currently manages that CMTS device.

Login Interactions with Assigned IP Subscribers

This section describes login interactions for assigned IP subscribers. In the example shown in Figure 12 on page 51, the subscriber activates a service through a portal. You could also have the subscriber activate a service through the Advanced Services Gateway.

Figure 12: Login Interactions with Assigned IP Subscribers



The sequence of events for logging in and creating sessions for assigned IP subscribers is:

1. The subscriber logs in to the portal.
2. The portal sends the subscriber's IP address to the NIC.
3. Based on the IP address, the NIC looks up the subscriber's SAE, CMTS device, and interface name, and returns this information to the portal.
4. The portal sends a getSubscriber message to the SAE. The message includes the subscriber's IP address, CMTS device, and interface name.
5. The SAE creates an assigned IP subscriber and performs a subscriber login. Specifically, it:
   a. Runs the interface classification script and creates a pseudointerface for the PCMM device driver.
      - If it finds a default policy, it pushes the policy to the CMTS device.
      - If it does not find a default policy, it continues with the next steps.
   b. Runs the subscriber classification script with the IP address of the subscriber. (Use the ASSIGNEDIP login type in subscriber classification scripts.)

c. Loads the subscriber profile.

d. Runs the subscriber authorization plug-ins.

e. Runs the subscriber tracking plug-ins.

f. Creates a subscriber session and stores the session data in the session store file.

6. The SAE pushes service policies for the subscriber session to the CMTS device.

Because the SAE is not notified when the subscriber logs out, the assigned IP idle timer begins when no service is active. The SAE removes the interface subscriber session when the timeout period ends.

Event Notification from an IP Address Manager

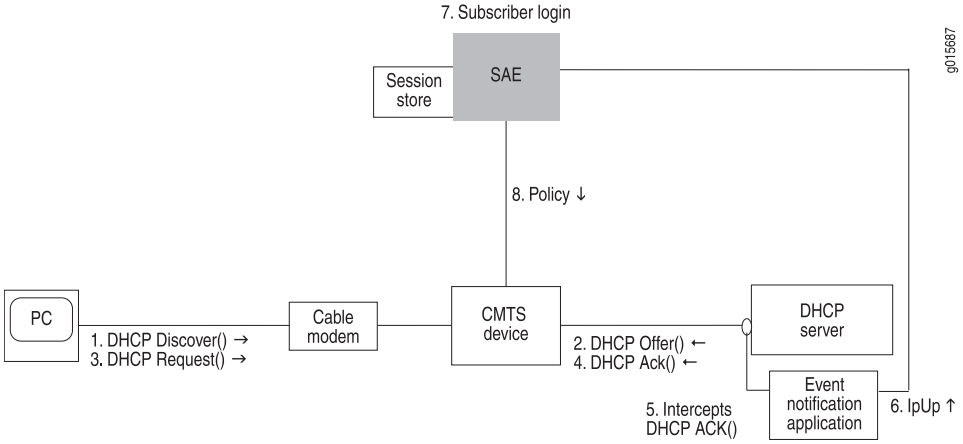
With the event notification method of logging in subscribers and creating subscriber sessions, the subscriber logs in to the CMTS device and obtains an IP address through an address server, usually a DHCP server. The SAE receives notifications about the subscriber, such as the subscriber's IP address, from an event notification application that is installed on the DHCP server.

To use this method of logging in subscribers, you can use the event notification API to create the application that notifies the SAE when events occur between the DHCP server and the CMTS device. You can also use Monitoring Agent, an application that was created with the event notification API, and that monitors DHCP or RADIUS messages for DHCP or RADIUS servers. See *SRC PE Sample Applications Guide*.

Login with Event Notification

This section describes login interactions using event notifications.

Figure 13: Login Interactions with Event Notification Application



The sequence of events for logging in subscribers and creating sessions is:

1. The DHCP client in the subscriber's computer sends a DHCP discover request to the DHCP server.
2. The DHCP server sends a DHCP offer to the subscriber's DHCP client.
3. The DHCP client sends a DHCP request to the DHCP server.
4. The DHCP server acknowledges the request by sending a DHCP Ack message to the DHCP client.
5. The event notification application that is running on the DHCP server intercepts the DHCP Ack message.
6. The event notification application sends an ipUp message to the SAE that notifies the SAE that an IP address is up.
7. The SAE performs a subscriber login. Specifically, it:
   a. Runs the interface classification script and creates a pseudointerface for the PCMM device driver.
      - If it finds a default policy, it pushes the policy to the CMTS device.
      - If it does not find a default policy, it continues with the next steps.
   b. Runs the subscriber classification script.
   c. Loads the subscriber profile.
   d. Runs the subscriber authorization plug-ins.
   e. Runs the subscriber tracking plug-ins.
   f. Creates a subscriber session and stores the session in the session store file.
8. The SAE provisions policies for the subscriber session on the CMTS device.

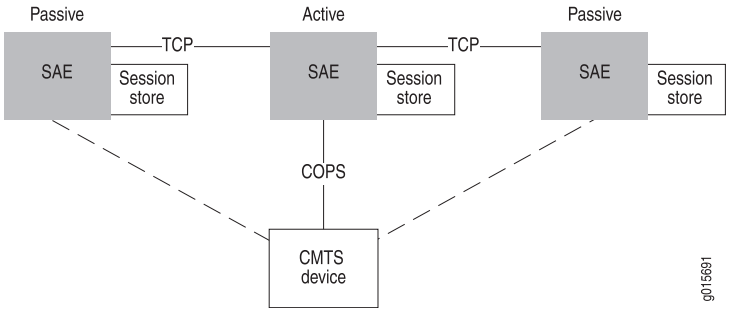
The ipUp event should be sent with a timeout set to the DHCP lease time. The DHCP server sends an ipUp event for each Ack sent to the client. The SAE restarts the timeout each time it receives an ipUp event.

If the client explicitly releases the DHCP address (that is, it sends a DHCP release event), the DHCP server sends an ipDown event. If the client does not renew the address, the lease expires on the DHCP server and the timeout expires on the SAE.

SAE Communities

For SAE redundancy in a cable network, you can have a community of two or more SAEs. SAEs in a community are given the role of either active SAE or passive SAE. The active SAE manages the connection to the CMTS device and keeps session data up to date within the community. Figure 14 on page 54 shows a typical SAE community.

Figure 14: SAE Community



When an SAE opens a connection to the CMTS device, it negotiates with other SAEs to determine which SAE controls the CMTS device. The SAE community manager and members of the community select the active SAE.

A passive SAE needs to take over as active SAE in any of the following cases:

- The active SAE shuts down or the connection between the CMTS device and the active SAE goes down. In this case, the active SAE notifies the passive SAEs, and one of the passive SAEs takes over as active SAE.
- A passive SAE does not receive a keepalive message from the active SAE within the keepalive interval. In this case, the passive SAE attempts to become the active SAE.

Storing Session Data

To aid in recovering from an SAE failover, the SAE stores subscriber and service session data. When the SAE manages a CMTS device, session data is stored locally in the SAE host's file system. The SRC component that controls the storage of session data on the SAE is called the session store. The session store queues data and then writes the data to session store files on the SAE host's disk. Once the data is written to disk, it can survive a server reboot.

For more information, see Fault Recovery.

PCMM Record-Keeping Server Plug-In

To allow the SAE's embedded policy server to communicate with a record-keeping server (RKS) in a PCMM environment, you need to use the PCMM record-keeping server plug-in. This plug-in is similar to the RADIUS accounting plug-ins, but it works with any RKS that is compliant with the PCMM specification. The RKS plug-in supports additional attributes: Application-Manager-ID, Request-Type, and Update-Reason. The plug-in sends all requests to the RKS as Acct-Status-Type=Interim-Update.

Related Topics

- Overview of a PCMM Environment on page 41
- Using the NIC Resolver in PCMM Environments on page 73
- Configuring the SAE to Manage PCMM Devices (SRC CLI) on page 58
- Initially Configuring the SAE

- Storing Subscriber and Service Session Data

CHAPTER 6

Configuring the SAE for a PCMM Environment (SRC CLI)

Configuring the SAE for a Cable Network Environment (SRC CLI)

The tasks to configure the SAE for a cable network environment are:

1. Configure the SAE to manage PCMM devices.

   "Configuring the SAE to Manage PCMM Devices (SRC CLI)" on page 58.

2. Configure the session store.

   See Configuring the Session Store Feature (SRC CLI).

3. Set up SAE communities.

   See "Setting Up SAE Communities (SRC CLI)" on page 61.

4. (Optional) Configure SAE properties for the event notification API.

   See "Configuring SAE Properties for the Event Notification API (SRC CLI)" on page 62 (if you are using an external address manager).

5. (Optional) Configure record-keeping server peers for plug-ins.

   See "Configuring Record-Keeping Server Peers for Plug-Ins (SRC CLI)" on page 63 (if you are using the RKS plug-in).

6. (Optional) Configure PCMM record-keeping server plug-ins.

See “Configuring PCMM Record-Keeping Server Plug-Ins (SRC CLI)” on page 64 (if you are using the SAE’s embedded policy server).

7. (Optional) Configure CMTS-specific RKS plug-ins.

   See “Configuring CMTS-Specific RKS Plug-Ins (SRC CLI)” on page 66.

In addition to configuring the SAE, you need to:

1. Configure the CMTS device in the directory (if you are using the SAE’s embedded policy server).

   See “Adding Objects for CMTS Devices (SRC CLI)” on page 69 .

2. Configure the NIC (if you are using assigned IP subscribers).

   See “Using the NIC Resolver” on page 102.

3. Enable the Common Open Policy Service (COPS) interface on the CMTS device. See the documentation for your CMTS device for information about how to do this.

Related Topics

- Overview of a PCMM Environment on page 41
- Configuring the SAE for a Cable Network Environment (C-Web Interface)
- Configuring the SAE to Manage PCMM Devices (C-Web Interface).
- Configuring an SAE Group .

Configuring the SAE to Manage PCMM Devices (SRC CLI)

The SAE connects to the PCMM device by using a COPS over TCP connection. The PCMM device driver controls this connection.

Use the following configuration statements to configure the SAE to manage CMTS devices:

```
shared sae configuration driver pcmm {
  keepalive-interval keepalive-interval ;
    tcp-connection-timeout tcp-connection-timeout ;
    application-manager-id application-manager-id ;
    message-timeout message-timeout ;
    cops-message-maximum-length cops-message-maximum-length ;
    cops-message-read-buffer-size cops-message-read-buffer-size ;
    cops-message-write-buffer-size cops-message-write-buffer-size ;
    sae-community-manager sae-community-manager ;
    disable-full-sync disable-full-sync ;
    disable-pcmm-i03-policy disable-pcmm-i03-policy ;
    session-recovery-retry-interval session-recovery-retry-interval ;
    element-id element-id ;
    default-rks-plug-in default-rks-plug-in ;
}
```

To configure the SAE to manage CMTS devices:

1. From configuration mode, access the configuration statement that configures the PCMM driver. In this sample procedure, the PCMM device driver is configured in the west-region group.

   user@host# edit shared sae group west-region configuration driver pcmm

2. Configure the interval between keepalive messages sent from the COPS client (the PCMM device) to the COPS server (the SAE).

   [edit shared sae group west-region configuration driver pcmm]
   user@host# set keepalive-interval *keepalive-interval*

3. Configure the timeout for opening a TCP connection to the PCMM device.

   [edit shared sae group west-region configuration driver pcmm]
   user@host# set tcp-connection-timeout *tcp-connection-timeout*

4. When this SAE is configured as the application manager, configure the identifier of the application manager.

   [edit shared sae group west-region configuration driver pcmm]
   user@host# set application-manager-id *application-manager-id*

5. Configure the time that the COPS server (the SAE) waits for a response to COPS requests from the COPS client (the PCMM device). Change this value only if a high number of COPS timeout events appear in the error log.

   [edit shared sae group west-region configuration driver pcmm]
   user@host# set message-timeout *message-timeout*

6. Configure the maximum length of a COPS message.

   [edit shared sae group west-region configuration driver pcmm]
   user@host# set cops-message-maximum-length *cops-message-maximum-length*

7. Configure the buffer size for receiving COPS messages from the COPS client.

   [edit shared sae group west-region configuration driver pcmm]
   user@host# set cops-message-read-buffer-size *cops-message-read-buffer-size*

8. Configure the buffer size for sending COPS messages to the COPS client.

   [edit shared sae group west-region configuration driver pcmm]
   user@host# set cops-message-write-buffer-size *cops-message-write-buffer-size*

9. Configure the name of the community manager that manages PCMM driver communities. Active SAEs are selected from this community.

   [edit shared sae group west-region configuration driver pcmm]
   user@host# set sae-community-manager *sae-community-manager*

10. Enable or disable state synchronization with PCMM policy servers.

    [edit shared sae group west-region configuration driver pcmm]
    user@host# set disable-full-sync *disable-full-sync*

11. Enable or disable the SAE to send classifiers to the router that comply with PCMM IO3. Disable this option if your network deployment has CMTS devices that do not support PCMM IO3.

    [edit shared sae group west-region configuration driver pcmm]
    user@host# set disable-pcmm-i03-policy ***disable-pcmm-i03-policy***

12. Configure the time between attempts by the SAE to restore service sessions that are being recovered in the background when state synchronization completes with a state-data-incomplete error.

    [edit shared sae group west-region configuration driver pcmm]
    user@host# set session-recovery-retry-interval ***session-recovery-retry-interval***

13. (Optional) Configure the unique identifier that the SAE uses to identify itself when it originates in record-keeping server (RKS) events.

    [edit shared sae group west-region configuration driver pcmm]
    user@host# set element-id ***element-id***

14. (Optional) Specify the name of the default RKS plug-in to which the SAE sends events for CMTS devices.

    [edit shared sae group west-region configuration driver pcmm]
    user@host# set default-rks-plug-in ***default-rks-plug-in***

15. (Optional) Verify your PCMM driver configuration.

```
[edit shared sae group west-region configuration driver pcmm]
user@host# show
keepalive-interval 45;
tcp-connection-timeout 5;
application-manager-id 1;
message-timeout 120000;
cops-message-maximum-length 204800;
cops-message-read-buffer-size 3000;
cops-message-write-buffer-size 3000;
sae-community-manager PcmmCommunityManager;
disable-full-sync true;
disable-pcmm-i03-policy true;
session-recovery-retry-interval 3600000;
element-id 1;
default-rks-plug-in rksTracking;
```

Related Topics

- Using the SAE in a PCMM Environment on page 50
- Connections to Managed Devices
- Configuring the SAE to Manage PCMM Devices (C-Web Interface)
- Configuring CMTS-Specific RKS Plug-Ins (SRC CLI) on page 66
- Initially Configuring the SAE

Setting Up SAE Communities (SRC CLI)

You can configure the following for SAE communities:

- Define the members of an SAE community by adding the IP addresses of SAEs in the community to the virtual router object of the network device in the directory.

  See "Creating Virtual Routers for the CMTS Device (SRC CLI)" on page 70 .

- Configure parameters for the SAE community manager.

  See "Configuring the SAE Community Manager" on page 61 .

- Specify the name of the community manager with the **set sae-community-manager** option in the PCMM driver configuration.

  See "Configuring the SAE to Manage PCMM Devices (SRC CLI)" on page 58.

- If there is a firewall in the network, configure the firewall to allow SAE messages through.

Related Topics

- Using the SAE in a PCMM Environment on page 50
- Setting Up SAE Communities (C-Web Interface)
- Initially Configuring the SAE
- Configuring SAE Properties for the Event Notification API (SRC CLI) on page 62

Configuring the SAE Community Manager

Use the following configuration statements to configure the SAE community manager that manages PCMM device communities:

```
shared sae configuration external-interface-features name CommunityManager {
  keepalive-interval keepalive-interval ;
    threads threads ;
    acquire-timeout acquire-timeout ;
    blackout-time blackout-time ;
}
```

To configure the community manager:

1. From configuration mode, access the configuration statements for the community manager. In this sample procedure, west_region is the name of the SAE group, and sae_mgr is the name of the community manager.

   ```
   user@host# edit shared sae group west-region configuration
     external-interface-features sae_mgr CommunityManager
   ```

2. Specify the interval between keepalive messages sent from the active SAE to the passive members of the community.

   ```
   [edit shared sae group west-region configuration external-interface-features sae_mgr
     CommunityManager]
   user@host# set keepalive-interval keepalive-interval
   ```

3. Specify the number of threads that are allocated to manage the community. You generally do not need to change this value.

   [edit shared sae group west-region configuration external-interface-features sae_mgr CommunityManager]
   user@host# set threads *threads*

4. Specify the amount of time an SAE waits for a remote member of the community when it is acquiring a distributed lock. You generally do not need to change this value.

   [edit shared sae group west-region configuration external-interface-features sae_mgr CommunityManager]
   user@host# set acquire-timeout *acquire-timeout*

5. Specify the amount of time that an active SAE must wait after it shuts down before it can try to become the active SAE of the community again.

   [edit shared sae group west-region configuration external-interface-features sae_mgr CommunityManager]
   user@host# set blackout-time *blackout-time*

6. (Optional) Verify the configuration of the SAE community manager.

```
[edit shared sae group west-region configuration
external-interface-features sae_mgr CommunityManager]
user@host# show
CommunityManager {
  keepalive-interval 30;
  threads 5;
  acquire-timeout 15;
  blackout-time 30;
}
```

Related Topics

- Using the SAE in a PCMM Environment on page 50
- Configuring the SAE Community Manager (C-Web Interface)
- Setting Up SAE Communities (SRC CLI) on page 61
- Initially Configuring the SAE .

Configuring SAE Properties for the Event Notification API (SRC CLI)

Use the following configuration statements to configure properties for the Event Notification API:

```
shared sae configuration external-interface-features name EventAPI {
  retry-time retry-time ;
    retry-limit retry-limit ;
    threads threads ;
}
```

To configure properties for the Event Notification API:

1. From configuration mode, access the configuration statements for the Event Notification API. In this sample procedure, west-region is the name of the SAE group, and event_api is the name of the Event API configuration.

   ```
   user@host# edit shared sae group west-region configuration
     external-interface-features event_api EventAPI
   ```

2. Specify the amount of time between attempts to send events that could not be delivered.

   ```
   [edit shared sae group west-region configuration external-interface-features event_api
     EventAPI]
   user@host# set retry-time retry-time
   ```

3. Specify the number of times an event fails to be delivered before the event is discarded.

   ```
   [edit shared sae group west-region configuration external-interface-features event_api
     EventAPII]
   user@host# set retry-limit retry-limit
   ```

4. Specify the number of threads allocated to process events.

   ```
   [edit shared sae group west-region configuration external-interface-features event_api
     EventAPI]
   user@host# set threads threads
   ```

5. (Optional) Verify the configuration of the Event Notification API properties.

   ```
   [edit shared sae group west-region configuration
   external-interface-features event_api EventAPI]
   user@host# show
   EventAPI {
     retry-time 300;
     retry-limit 5;
     threads 5;
   }
   ```

Related Topics

- Using the SAE in a PCMM Environment on page 50
- Configuring SAE Properties for the Event Notification API (C-Web Interface)
- Initially Configuring the SAE
- Configuring the SAE to Manage PCMM Devices (SRC CLI) on page 58

Configuring Record-Keeping Server Peers for Plug-Ins (SRC CLI)

An RKS peer is an instance of an RKS. A PCMM environment has a primary RKS and optionally a secondary RKS. The primary RKS is mandatory, and you assign the RKS as primary by configuring it as the default peer in the RKS plug-in. The secondary RKS is optional, and it is an RKS peer that is not configured as the default peer. If you define multiple nondefault peers, one of them is randomly chosen to be the secondary RKS.

RKS peers are configured in the peer group for each PCMM RKS plug-in instance. To create an RKS peer group:

Use the following configuration statements to configure an RKS peer group.

```
shared sae configuration plug-ins name name pcmm-rks peer-group name {
  server-address server-address ;
    server-port server-port ;
}
```

To configure an RKS peer group:

1. From configuration mode, access the configuration statements for RKS plug-ins. In this sample procedure, west-region is the name of the SAE group, and rksPlugin is the name of the plug-in and rksPeer is the name of the peer group.

   ```
   user@host# edit shared sae group west-region configuration plug-ins name rksPlugin
     pcmm-rks peer-group rksPeer
   ```

2. Specify the IP address of the RKS server to which the SAE sends accounting data.

   ```
   [edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks
     peer-group rksPeer]
   user@host# set server-address server-address
   ```

3. Specify the port used for sending accounting packets.

   ```
   [edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks
     peer-group rksPeer]
   user@host# set server-port server-port
   ```

4. (Optional) Verify your configuration.

   ```
   [edit shared sae group west-region configuration plug-ins name rksPlugin
    pcmm-rks peer-group rksPeer]
   user@host# show
   server-address 10.10.3.60;
   server-port 1812;
   ```

Related Topics

- Using the SAE in a PCMM Environment on page 50
- Configuring Record-Keeping Server Peers for Plug-Ins (C-Web Interface)
- Configuring PCMM Record-Keeping Server Plug-Ins (SRC CLI) on page 64
- Configuring CMTS-Specific RKS Plug-Ins (SRC CLI) on page 66
- Initially Configuring the SAE

Configuring PCMM Record-Keeping Server Plug-Ins (SRC CLI)

Use the following configuration statements to configure an RKS plug-in.

```
shared sae configuration plug-ins name name pcmm-rks {
  load-balancing-mode (failover | roundRobin);
```

```
    failback-timer failback-timer;
    retry-interval retry-interval ;
    maximum-queue-length maximum-queue-length ;
    bind-address bind-address ;
    udp-port udp-port ;
    feid-mso-data feid-mso-data ;
    feid-mso-domain-name feid-mso-domain-name ;
    trusted-element;
    default-peer default-peer ;
}
```

To configure an RKS plug-in:

1. From configuration mode, access the configuration statements for RKS plug-ins. In this sample procedure, west-region is the name of the SAE group, and rksPlugin is the name of the plug-in.

   ```
   user@host# edit shared sae group west-region configuration plug-ins name rksPlugin
     pcmm-rks
   ```

2. Specify the mode for load-balancing RKSs.

   ```
   [edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]
   user@host# set load-balancing-mode (failover | roundRobin)
   ```

3. Specify if and when the SAE attempts to fail back to the default peer.

   ```
   [edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]
   user@host# set failback-timer failback-timer
   ```

4. Specify the time the SAE waits for a response from an RKS before it resends the packet.

   ```
   [edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]
   user@host# set retry-interval retry-interval
   ```

5. Specify the maximum number of unacknowledged messages that the plug-in receives from the RKS before it discards new messages.

   ```
   [edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]
   user@host# set maximum-queue-length maximum-queue-length
   ```

6. (Optional) Specify the source IP address that the plug-in uses to communicate with the RKS.

   ```
   [edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]
   user@host# set bind-address bind-address
   ```

7. (Optional) Specify the source UDP port or a pool of ports that the plug-in uses to communicate with the RKS.

   ```
   [edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]
   user@host# set udp-port udp-port
   ```

8. (Optional) Specify the multiple service operator (MSO)—defined data in the financial entity ID (FEID) attribute, which is included in event messages.

   [edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]
   user@host# set feid-mso-data ***feid-mso-data***

9. (Optional) Specify the MSO domain name in the FEID attribute that uniquely identifies the MSO for billing and settlement purposes.

   [edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]
   user@host# set feid-mso-domain-name ***feid-mso-domain-name***

10. (Optional) When the SAE is running as a policy server—which means that the SAE sends event messages directly to the RKS—enable the SAE as a trusted network element.

    [edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]
    user@host# set trusted-element

11. Specify the name of the primary RKS peer to which the SAE sends accounting packets.

    See “Configuring Record-Keeping Server Peers for Plug-Ins (SRC CLI)” on page 63.

    [edit shared sae group west-region configuration plug-ins name rksPlugin pcmm-rks]
    user@host# set default-peer ***default-peer***

12. (Optional) Verify your RKS plug-in configuration.

```
[edit shared sae group west-region configuration plug-ins name rksPlugin
 pcmm-rks]
user@host> show
load-balancing-mode failover;
failback-timer -1;
retry-interval 3000;
maximum-queue-length 10000;
feid-mso-domain-name abcd.com;
trusted-element;
default-peer radius01;
```

13. (Optional) Specify an RKS plug-in for specific CMTS devices.

    See “Configuring CMTS-Specific RKS Plug-Ins (SRC CLI)” on page 66 .

Related Topics

- Using the SAE in a PCMM Environment on page 50
- Overview of a PCMM Environment on page 41
- Configuring PCMM Record-Keeping Server Plug-Ins (C-Web Interface)
- Initially Configuring the SAE

Configuring CMTS-Specific RKS Plug-Ins (SRC CLI)

You can configure an RKS plug-in for specific CMTS devices. When there are events for the CMTS device, the SAE sends the events to the specified plug-in.

Use the following configuration statement to assign a CMTS-specific RKS plug-in.

```
shared sae configuration driver pcmm cmts-specific-rks-plug-ins name {
  rks-plug-in rks-plug-in ;
}
```

To configure a CMTS-specific RKS plug-in:

1. From configuration mode, access the configuration statements for RKS plug-ins. In this sample procedure, west-region is the name of the SAE group, and cmtsPlugin is the name of the plug-in assignment.

   ```
   user@host# edit shared sae group west-region configuration driver pcmm
     cmts-specific-rks-plug-ins cmtsPlugin
   ```

2. Specify the name of the CMTS-specific RKS plug-in.

   ```
   [edit shared sae group west-region configuration driver pcmm
     cmts-specific-rks-plug-ins cmtsPlugin]
   user@host# set rks-plug-in rks-plug-in
   ```

3. (Optional) Verify your configuration.

   ```
   [edit shared sae group west-region configuration driver pcmm
   cmts-specific-rks-plug-ins cmtsPlugin]
   user@host# show
   rks-plug-in rksPlugin;
   ```

Related Topics

- Configuring CMTS-Specific RKS Plug-Ins (C-Web Interface)
- Configuring Record-Keeping Server Peers for Plug-Ins (SRC CLI) on page 63
- Configuring PCMM Record-Keeping Server Plug-Ins (SRC CLI) on page 64
- Adding Objects for CMTS Devices (SRC CLI) on page 69
- Initially Configuring the SAE

CHAPTER 7

Adding Objects for CMTS Devices (SRC CLI)

Adding Objects for CMTS Devices (SRC CLI)

To manage CMTS devices, the SAE creates and manages pseudointerfaces that it associates with a virtual router object. Each CMTS device in the SRC network must appear in the configuration as a router object, and it must be associated with a virtual router object called default. The router and virtual router are not actually configured on the CMTS device; the router and virtual router provide a way for the SAE to manage the CMTS device by using the SAE's embedded policy server.

Use the following configuration statements to add a router object:

```
shared network device name {
  description description ;
    management-address management-address ;
    device-type (junose | junos | pcmm | proxy);
    qos-profile [ qos-profile ...];
}
```

To add a router:

1. From configuration mode, access the configuration statements that configure network devices. You must specify the name of a device with lowercase characters. In this sample procedure, pcmm_dtr is the name of the object.

   ```
   user@host# edit shared network device pcmm_dtr
   ```

2. (Optional) Add a description for the CMTS device.

   ```
   [edit shared network device pcmm_dtr]
   user@host#  set description description
   ```

3. Add the IP address of the CMTS device.

   ```
   [edit shared network device pcmm_dtr]
   user@host#  set management-address management-address
   ```

4. (Optional) Specify the type of device that you are adding.

```
[edit shared network device pcmm_dtr]
user@host#  set device-type pcmm
```

5. (Optional) Verify your configuration.

```
[edit shared network device pcmm_dtr]
user@host# show
description "CMTS device";
management-address 192.168.3.5;
device-type pcmm;
interface-classifier {
  rule rule-0 {
    script #;
  }
}
```

Related Topics

- Connections to Managed Devices
- Configuring CMTS-Specific RKS Plug-Ins (SRC CLI) on page 66
- Creating Virtual Routers for the CMTS Device (SRC CLI) on page 70

Creating Virtual Routers for the CMTS Device (SRC CLI)

You need to add a virtual router object called default to the CMTS device.

Use the following configuration statements to add a virtual router:

```
shared network device name virtual-router name {
  sae-connection [ sae-connection ...];
    snmp-read-community snmp-read-community ;
    snmp-write-community snmp-write-community ;
    scope [ scope ...];
    local-address-pools local-address-pools ;
    static-address-pools static-address-pools ;
    tracking-plug-in [ tracking-plug-in ...];
}
```

To add a virtual router:

1. From configuration mode, access the configuration statements for virtual routers. You must specify the name of a device with lowercase characters. In this sample procedure, pcmm_dtr is the name of the router and default is the name of the virtual router.

```
user@host# edit shared network device pcmm_dtr virtual-router default
```

2. Specify the addresses of SAEs that can manage this router. This step is required for the SAE to work with the router.

```
[edit shared network device pcmm_dtr virtual-router default]
user@host# set sae-connection [ sae-connection ...]
```

To specify the active SAE and the redundant SAE, enter an exclamation point (!) after the hostname or IP address of the connected SAE. For example:

```
[edit shared network device pcmm_dtr virtual-router default]
user@host# set sae-connection [sae1! sae2!]
```

3. (Optional) Specify an SNMP community name for SNMP read-only operations for this VR.

```
[edit shared network device pcmm_dtr virtual-router default]
user@host# set snmp-read-community snmp-read-community
```

4. (Optional) Specify an SNMP community name for SNMP write operations for this virtual router.

```
[edit shared network device pcmm_dtr virtual-router default]
user@host# set snmp-write-community snmp-write-community
```

5. (Optional) Specify service scopes assigned to this virtual router.

   See Configuring Service Scopes (SRC CLI).

```
[edit shared network device pcmm_dtr virtual-router default]
user@host# set scope [ scope ...]
```

6. (Optional) Specify the list of IP address pools that a CMTS virtual router currently manages and stores.

   If you are using assigned IP subscribers along with the network information collector (NIC), you need to configure either a local or static address pool so that the NIC can resolve the IP-to-SAE mapping.

```
[edit shared network device pcmm_dtr virtual-router default]
user@host# set local-address-pools local-address-pools
```

7. (Optional) Specify the list of IP address pools that a CMTS VR manages but does not store.

   If you are using assigned IP subscribers along with the NIC, you need to configure either a local or static address pool so that the NIC can resolve the IP-to-SAE mapping.

```
[edit shared network device pcmm_dtr virtual-router default]
user@host# set static-address-pools static-address-pools
```

8. (Optional) Specify the plug-ins that track interfaces that the SAE manages on this virtual router.

```
[edit shared network device pcmm_dtr virtual-router default]
user@host# tracking-plug-in [ tracking-plug-in ...]
```

9. (Optional) Verify your configuration.

```
[edit shared network device pcmm_dtr virtual-router default]
user@host# show
```

```
sae-connection [ 10.14.39.2 10.10.5.30 ];
snmp-read-community ********;
snmp-write-community ********;
scope POP-Westford;
local-address-pools "10.25.8.0 10.25.20.255";
tracking-plug-in rksPlugin;
```

Related Topics

- Adding Objects for CMTS Devices (SRC CLI) on page 69
- Configuring CMTS-Specific RKS Plug-Ins (SRC CLI) on page 66
- Associating Security Names with a Community (SRC CLI)

CHAPTER 8

Using the NIC Resolver in a PCMM Environment

Using the NIC Resolver in PCMM Environments

If you are using the NIC to map the subscriber IP address to the SAE, you need to configure a NIC host. The NIC system uses IP address pools to map IP addresses to SAEs. You configure the local address pools in the application manager configuration for a policy server group. These pools are published in the NIC. The NIC maps subscriber IP addresses in requests received through the portal or Advanced Services Gateway to the policy server group that currently manages that CMTS device.

The OnePopPcmm sample configuration data supports this scenario for a PCMM environment in which you use the assigned IP subscriber method to log in subscribers and in which you use the NIC to determine the subscriber's SAE. The OnePopPcmm configuration supports one point of presence (POP). NIC replication can be used to provide high availability. The realm for this configuration accommodates the situation in which IP pools are configured locally on each application manager group object.

The resolution process takes a subscriber's IP address as the key and returns a reference to the SAE managing this subscriber as the value.

The following agents collect information for resolvers in this realm:

- Directory agent PoolVr collects and publishes information about the mappings of IP pools to the policy server group.
- Directory agent VrSaeId collects and publishes information about the mappings of policy server groups to SAEs.

Related Topics

- Overview of a PCMM Environment on page 41
- Using the SAE in a PCMM Environment on page 50
- Specifying Application Manager Identifiers for Policy Servers (C-Web Interface)
- Configuring the NIC (SRC CLI)
- OnePopPcmm Scenario

CHAPTER 9

Using PCMM Policy Servers

Overview of the JPS

In a PCMM environment, the policy server acts as a policy decision point (PDP) and policy enforcement point (PEP) that manages the relationships between application managers and cable management termination system (CMTS) devices.

The JPS is a PCMM-compliant policy server. The JPS must be deployed in an SRC environment that satisfies these conditions:

- Organizes PCMM devices into groups (for example, one or more per POP). For redundancy, a community of two or more JPSs will manage each group of PCMM devices.
- Achieves successful state synchronization by requiring an application manager (for example, a pair of redundant SAEs) to talk to one JPS instance at a time.
- Uses IPSec connections for the network interfaces.

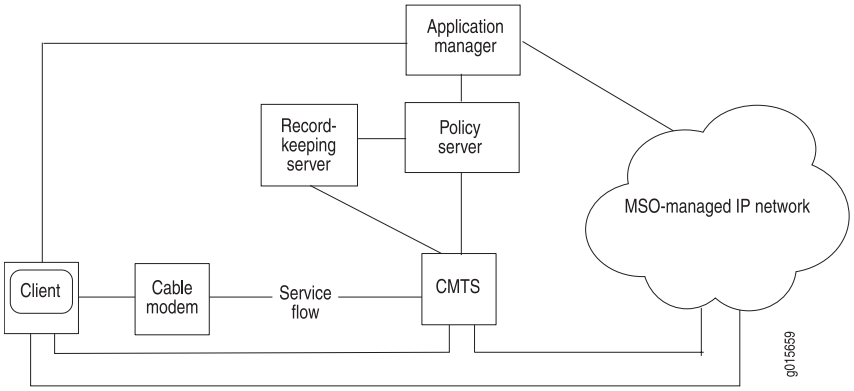
Related Topics

- JPS Framework on page 75
- JPS Interfaces on page 76
- Starting the JPS (SRC CLI) on page 103
- Configuring the JPS (SRC CLI) on page 81
- Monitoring the JPS on page 105

JPS Framework

Figure 15 on page 76 depicts the PCMM architectural framework. The JPS communicates with application managers, CMTS devices, and record-keeping servers.

Figure 15: PCMM Architectural Framework



The interactions between the various PCMM components are centered on the policy server. In the PCMM architecture, these basic interactions occur:

1. A client requests a multimedia service from an application manager.
2. Depending on the client type and its QoS signaling capabilities, the application manager relays the request to a policy server.
3. The policy server relays the request to the CMTS device and is responsible for provisioning the policies on a CMTS device.

   Depending on the request, the policy server records an event for the policy request and provides that information to the record-keeping server (RKS).
4. The CMTS device performs admission control and manages network resources through Data over Cable Service Interface Specifications (DOCSIS) service flows based on the provisioned policies.
5. The RKS receives event messages from other network elements, such as the policy server or CMTS device, and acts as a short-term repository for the messages.

Related Topics

- Overview of the JPS on page 75
- JPS Interfaces on page 76
- Policy Information Model
- Configuring the JPS (SRC CLI) on page 81

JPS Interfaces

The JPS has interfaces, implemented as plug-ins, to communicate with:

- Application managers, such as the SAE
- Record-keeping servers
- CMTS devices

The JPS is relatively stateless, but the individual plug-ins can be stateful.

The JPS uses the Common Open Policy Service (COPS) protocol as specified in the PacketCable Multimedia Specification PKT-SP-MM-I03-051221 for its interface connections. The JPS communicates with the CMTS device and the application manager by using a COPS over Transmission Control Protocol (TCP) connection.

Application Manager to Policy Server Interface

To allow the JPS to communicate with the application manager, this interface accepts and manages COPS over TCP connections from application managers, such as the SAE.

Policy Server to RKS Interface

To allow the JPS to communicate with a set of redundant record-keeping servers, this interface sends a policy event message to the RKS when receiving a PCMM-COPS gate control (request, delete, update) message. This interface also sends time change events to the RKS.

Policy Server to CMTS Interface

To allow the JPS to communicate with policy enforcement points (PCMM devices), this interface initiates and manages COPS over TCP connections with CMTS devices.

Related Topics

- Overview of the JPS on page 75
- JPS Framework on page 75
- Configuring the JPS (SRC CLI) on page 81
- Connections to Managed Devices
- Viewing JPS State on page 106

CHAPTER 10

Configuring the JPS (SRC CLI)

Configuration Statements for the JPS

Use the following configuration statements to configure the JPS at the **[edit]** hierarchy level.

slot *number* jps {

```
    java-heap-size java-heap-size;
    snmp-agent;
    policy-server-id policy-server-id;
    use-psid-in-gate-commands;
    cmts-message-buffer-size cmts-message-buffer-size;
    am-message-buffer-size am-message-buffer-size;
}
slot number jps am-interface {
    pep-id pep-id;
    listening-address listening-address;
    validate-pcmm-objects;
    message-max-length message-max-length;
    message-read-buffer-size message-read-buffer-size;
    message-write-buffer-size message-write-buffer-size;
    open-connection-timeout open-connection-timeout;
}
slot number jps cmts-interface {
    cmts-addresses [cmts-addresses...];
    keepalive-interval keepalive-interval;
    synch-despite-unreachable-pep;
    synch-despite-pre-i03-pep;
    use-ssq-ssc-with-pre-i03-pep;
    local-address local-address;
    message-max-length message-max-length;
    message-read-buffer-size message-read-buffer-size;
    message-write-buffer-size message-write-buffer-size;
    open-connection-timeout open-connection-timeout;
    connection-open-retry-interval connection-open-retry-interval;
    sent-message-timeout sent-message-timeout;
    validate-pcmm-objects;
}
slot number jps cmts-registry cmts cmts-ip ...
slot number jps cmts-registry cmts cmts-ip range-pool pool-index {
    low low;
    high high;
}
slot number jps cmts-registry cmts cmts-ip subnet-pool subnet {
    exclude [exclude];
}
slot number jps logger name ...
slot number jps logger name file {
    filter filter;
    filename filename;
    rollover-filename rollover-filename;
    maximum-file-size maximum-file-size;
}
slot number jps logger name syslog {
    filter filter;
    host host;
    facility facility;
    format format;
}
slot number jps rks-interface {
    element-id element-id;
    local-address local-address;
    local-port local-port;
```



```
    retry-interval retry-interval;
    local-timeout local-timeout;
    mso-data mso-data;
    mso-domain-name mso-domain-name;
    default-rks-pair default-rks-pair;
    pending-rks-event-max-size pending-rks-event-max-size;
    pending-rks-event-max-age pending-rks-event-max-age;
    held-decs-max-size held-decs-max-size;
    held-decs-max-age held-decs-max-age;
    bcid-cache-size bcid-cache-size;
    bcid-cache-age bcid-cache-age;
    use-default-when-am-requests-unconfigured-rks;
}
slot number jps rks-interface am am-name {
    am-id am-id;
    rks-pair-name rks-pair-name;
    trusted;
}
slot number jps rks-interface rks-pair rks-pair-name {
    primary-address primary-address;
    primary-port primary-port;
    secondary-address secondary-address;
    secondary-port secondary-port;
}
```

Related Topics

- Overview of the JPS on page 75
- JPS Interfaces on page 76
- Configuring the JPS (SRC CLI) on page 81
- For detailed information about each configuration statement, see the *SRC PE CLI Command Reference*

Configuring the JPS (SRC CLI)

You can modify the JPS configuration, which includes configuring the logging destinations and connections to the JPS interfaces. Any configuration changes will be applied within 15 seconds.

Before you configure the JPS, deploy an SRC-managed PCMM network. For more information about PCMM and the SRC software, see “Overview of a PCMM Environment” on page 41.

You can configure the subscriber configuration, which maps a subscriber address to the CMTS address.

The tasks to configure the JPS for a cable network environment are:

- Modifying the JPS Configuration (SRC CLI) on page 82
- Modifying the Subscriber Configuration (SRC CLI) on page 95

In addition to configuring the JPS, you might need to perform these tasks:

- Configuring the SAE to Interact with the JPS (SRC CLI) on page 97
- Using the NIC Resolver on page 102

Related Topics

- Overview of the JPS on page 75
- Configuring the JPS (C-Web Interface)
- Configuring General Properties for the JPS (SRC CLI) on page 82
- Monitoring the JPS on page 105
- Configuration Statements for the JPS on page 79

Modifying the JPS Configuration (SRC CLI)

To modify the current JPS configuration:

1. Configure general properties for the JPS, including Java heap memory, maximum number of buffered messages for CMTS and application manager destinations, and policy server identifiers.

   See “Configuring General Properties for the JPS (SRC CLI)” on page 82.

   See “Specifying Policy Server Identifiers in Messages (SRC CLI)” on page 83.
2. Configure logging destinations for the JPS.

   See “Configuring Logging Destinations for the JPS (SRC CLI)” on page 84.
3. Configure the connections to the JPS interfaces.

   See “Specifying Connections to the Application Managers (SRC CLI)” on page 86.

   See “Specifying Connections to CMTS Devices (SRC CLI)” on page 92.

Related Topics

- Overview of the JPS on page 75
- Modifying the JPS Configuration (C-Web Interface)
- Configuring the JPS (SRC CLI) on page 81
- Viewing JPS State on page 106

Configuring General Properties for the JPS (SRC CLI)

Use the following configuration statements to configure general properties for the JPS:

```
slot number jps {
  java-heap-size java-heap-size;
  snmp-agent;
  cmts-message-buffer-size cmts-message-buffer-size;
  am-message-buffer-size am-message-buffer-size;
}
```

To configure general properties for the JPS:

1. From configuration mode, access the configuration statement that configures the general properties.

   ```
   user@host# edit slot 0 jps
   ```

2. (Optional) Specify the maximum amount of memory available to the Java Runtime Environment (JRE).

   ```
   [edit slot 0 jps]
   user@host# set java-heap-size java-heap-size
   ```

3. (Optional) Enable the JPS to communicate with the SNMP agent.

   ```
   [edit slot 0 jps]
   user@host# set snmp-agent
   ```

4. (Optional) Specify the maximum number of messages buffered for each CMTS destination.

   ```
   [edit slot 0 jps]
   user@host# set cmts-message-buffer-size cmts-message-buffer-size
   ```

5. (Optional) Specify the maximum number of messages buffered for each application manager destination.

   ```
   [edit slot 0 jps]
   user@host# set am-message-buffer-size am-message-buffer-size
   ```

6. (Optional) Verify your configuration.

   ```
   [edit slot 0 jps]
   user@host# show
   ```

Related Topics

- Overview of the JPS on page 75
- Modifying the JPS Configuration (C-Web Interface)
- Configuring the JPS (SRC CLI) on page 81
- Configuring Logging Destinations for the JPS (SRC CLI) on page 84

Specifying Policy Server Identifiers in Messages (SRC CLI)

Use the following configuration statements to configure policy server identifiers for the JPS:

```
slot number jps {
    policy-server-id policy-server-id;
    use-psid-in-gate-commands;
}
```

To configure policy server identifiers for the JPS:

1. From configuration mode, access the configuration statement that configures the policy server identifiers.

   ```
   user@host# edit slot 0 jps
   ```

2. (Optional) Specify the policy server identifier so that the JPS can be identified in messages sent to CMTS devices.

   ```
   [edit slot 0 jps]
   user@host# set policy-server-id policy-server-id
   ```

3. (Optional) Configure the JPS so that the policy server identifier is specified in messages sent to the RKS.

   ```
   [edit slot 0 jps]
   user@host# set use-psid-in-gate-commands
   ```

   When the JPS is communicating only with PCMM I03 CMTS devices, the value must be true. When the JPS is communicating with any pre-PCMM I03 CMTS devices, the value must be false.

4. (Optional) Verify your configuration.

   ```
   [edit slot 0 jps]
   user@host# show
   ```

Related Topics

- Modifying the JPS Configuration (C-Web Interface)
- Specifying Application Manager Identifiers for Policy Servers (SRC CLI) on page 99
- Adding Objects for Policy Servers to the Directory (SRC CLI) on page 100
- Modifying the JPS Configuration (SRC CLI) on page 82
- Viewing Server Process Information on page 105

Configuring Logging Destinations for the JPS (SRC CLI)

By default, the JPS has four logging destinations.

Use the following configuration statements to configure logging destinations for the JPS:

```
slot number jps logger name ...
slot number jps logger name file {
    filter filter;
    filename filename;
    rollover-filename rollover-filename;
    maximum-file-size maximum-file-size;
}
slot number jps logger name syslog {
    filter filter;
    host host;
    facility facility;
    format format;
}
```

Related Topics

- Overview of the JPS on page 75
- Modifying the JPS Configuration (C-Web Interface)
- Configuring JPS to Store Log Messages in a File (SRC CLI) on page 85
- Configuring JPS to Send Log Messages to System Logging Facility (SRC CLI) on page 85

Configuring JPS to Store Log Messages in a File (SRC CLI)

To configure logging destinations to store log messages in a file:

1. From configuration mode, access the configuration statement that configures the name and type of logging destination. In this sample procedure, the logging destination called log2 is configured.

   user@host# edit slot 0 jps logger log2 file

2. Specify the properties for the logging destination.

   [edit slot 0 jps logger log2 file]
   user@host# set ?

   For more information about configuring properties for the logging destination, see Overview of Logging for SRC Components.

3. (Optional) Verify your configuration.

```
[edit slot 0 jps logger log2]
user@host# show
file {
  filter !NoAckRksEvent,/info-;
  filename var/log/jps_info.log;
  rollover-filename var/log/jps_info.alt;
  maximum-file-size 2000000000;
}
```

Related Topics

- Overview of the JPS on page 75
- Configuring the JPS (SRC CLI) on page 81
- Configuring Logging Destinations for the JPS (SRC CLI) on page 84
- Configuring JPS to Send Log Messages to System Logging Facility (SRC CLI) on page 85

Configuring JPS to Send Log Messages to System Logging Facility (SRC CLI)

To configure logging destinations to send log messages to the system logging facility:

1. From configuration mode, access the configuration statement that configures the name and type of logging destination. In this sample procedure, the logging destination called log5 is configured.

   user@host# edit slot 0 jps logger log5 syslog

2. Specify the properties for the logging destination.

   ```
   [edit slot 0 jps logger log5 syslog]
   user@host# set ?
   ```

   For more information about configuring properties for the logging destination, see Overview of Logging for SRC Components.

3. (Optional) Verify your configuration.

   ```
   [edit slot 0 jps logger log5]
   user@host# show
   ```

Related Topics

- Overview of the JPS on page 75
- Configuring the JPS (SRC CLI) on page 81
- Configuring Logging Destinations for the JPS (SRC CLI) on page 84
- Configuring JPS to Store Log Messages in a File (SRC CLI) on page 85

Specifying Connections to the Application Managers (SRC CLI)

Use the following configuration statement to configure the application manager–to–policy server interface (PKT-MM3) so that the policy server can communicate with application managers:

```
slot number jps am-interface {
    pep-id pep-id;
    listening-address listening-address;
    validate-pcmm-objects;
    message-max-length message-max-length;
    message-read-buffer-size message-read-buffer-size;
    message-write-buffer-size message-write-buffer-size;
    open-connection-timeout open-connection-timeout;
}
```

To configure the connections to the application managers:

1. From configuration mode, access the configuration statement that configures the application manager–to–policy server interface.

   ```
   user@host# edit slot 0 jps am-interface
   ```

2. (Optional) Specify the network-wide unique identifier for this JPS instance.

   ```
   [edit slot 0 jps am-interface]
   user@host# set pep-id pep-id
   ```

   Changes apply only to COPS connections that are established after you make the change.

3. (Optional) Specify the local IP address on which the JPS listens for incoming connections from application managers.

```
[edit slot 0 jps am-interface]
user@host# set listening-address listening-address
```

Changes take effect only after you restart the JPS (see "Restarting the JPS (SRC CLI)" on page 103).

4. (Optional) Specify whether to validate PCMM objects received from PDPs.

```
[edit slot 0 jps am-interface]
user@host# set validate-pcmm-objects
```

5. (Optional) Specify the maximum length of incoming messages.

```
[edit slot 0 jps am-interface]
user@host# set message-max-length message-max-length
```

6. (Optional) Specify the size of message read buffer.

```
[edit slot 0 jps am-interface]
user@host# set message-read-buffer-size message-read-buffer-size
```

7. (Optional) Specify the size of message write buffer.

```
[edit slot 0 jps am-interface]
user@host# set message-write-buffer-size message-write-buffer-size
```

8. (Optional) Specify the maximum time to wait for the initial PCMM messages to be exchanged after a TCP connection is established.

```
[edit slot 0 jps am-interface]
user@host# set open-connection-timeout open-connection-timeout
```

The connection is dropped when initial PCMM messages are not exchanged within this time period.

9. (Optional) Verify your configuration.

```
[edit slot 0 jps am-interface]
user@host# show
pep-id SDX-JPS;
listening-address ;
validate-pcmm-objects;
message-max-length 204800;
message-read-buffer-size 1000000;
message-write-buffer-size 1000000;
open-connection-timeout 5;
```

Related Topics

- Modifying the JPS Configuration (C-Web Interface)
- Specifying Connections to CMTS Devices (SRC CLI) on page 92
- Modifying the JPS Configuration (SRC CLI) on page 82
- Specifying Application Managers for the Policy Server (SRC CLI) on page 98
- Viewing JPS State on page 106

Configuring Connections to RKSs (SRC CLI)

To configure connections to RKSs:

Specifying Connections to RKSs (SRC CLI)

To configure the policy server-to-RKS interface (PKT-MM4) so that policy events can be sent to the RKS, you can configure RKS pairs (see "Configuring RKS Pairs (SRC CLI)" on page 90) and their associated application managers (see "Configuring RKS Pairs for Associated Application Managers (SRC CLI)" on page 91).

Use the following configuration statement to configure the policy server-to-RKS interface:

```
slot number jps rks-interface {
    element-id element-id;
    local-address local-address;
    local-port local-port;
    retry-interval retry-interval;
    local-timeout local-timeout;
    mso-data mso-data;
    mso-domain-name mso-domain-name;
    default-rks-pair default-rks-pair;
    pending-rks-event-max-size pending-rks-event-max-size;
    pending-rks-event-max-age pending-rks-event-max-age;
    held-decs-max-size held-decs-max-size;
    held-decs-max-age held-decs-max-age;
    bcid-cache-size bcid-cache-size;
    bcid-cache-age bcid-cache-age;
    use-default-when-am-requests-unconfigured-rks;
}
```

To configure the policy server-to-RKS interface:

1. From configuration mode, access the configuration statement that configures the policy server-to-RKS interface.

   ```
   user@host# edit slot 0 jps rks-interface
   ```

2. ier for RKS event origin.

   ```
   [edit slot 0 jps rks-interface]
   user@host# set element-id element-id
   ```

3. (Optional) Specify the source IP address that the plug-in uses to communicate with the RKS.

   ```
   [edit slot 0 jps rks-interface]
   user@host# set local-address local-address
   ```

   If no value is specified and there is more than one local address, the JPS randomly selects a local address to be used as the source address.

4. (Optional) Specify the source UDP port or a pool of ports that the plug-in uses to communicate with the RKS.

   ```
   [edit slot 0 jps rks-interface]
   user@host# set local-port local-port
   ```

5. (Optional) Specify the time the JPS waits for a response from an RKS before it resends the packet.

   ```
   [edit slot 0 jps rks-interface]
   user@host# set retry-interval retry-interval
   ```

   The JPS keeps sending packets until either the RKS acknowledges the packet or the maximum timeout is reached.

6. (Optional) Specify the maximum time the JPS waits for a response from an RKS.

   ```
   [edit slot 0 jps rks-interface]
   user@host# set local-timeout local-timeout
   ```

7. (Optional) Specify the MSO-defined data in the financial entity ID (FEID) attribute, which is included in event messages.

   ```
   [edit slot 0 jps rks-interface]
   user@host# set mso-data mso-data
   ```

8. (Optional) Specify the MSO domain name in the FEID attribute that uniquely identifies the MSO for billing and settlement purposes.

   ```
   [edit slot 0 jps rks-interface]
   user@host# set mso-domain-name mso-domain-name
   ```

9. (Optional) Specify the default RKS pair that the JPS uses unless an RKS pair is configured for a given application manager.

   ```
   [edit slot 0 jps rks-interface]
   user@host# set default-rks-pair default-rks-pair
   ```

10. (Optional) Specify the maximum number of RKS events waiting for Gate-Set-Ack, Gate-Set-Err, Gate-Del-Ack, and Gate-Del-Err messages.

    ```
    [edit slot 0 jps rks-interface]
    user@host# set pending-rks-event-max-size pending-rks-event-max-size
    ```

11. (Optional) Specify the oldest age of RKS events waiting for Gate-Set-Ack, Gate-Set-Err, Gate-Del-Ack, and Gate-Del-Err messages.

    ```
    [edit slot 0 jps rks-interface]
    user@host# set pending-rks-event-max-age pending-rks-event-max-age
    ```

    The maximum age must be greater than sent-message-timeout of the corresponding CMTS interface.

12. (Optional) Specify the maximum number of outstanding Gate-Info requests.

```
[edit slot 0 jps rks-interface]
user@host# set held-decs-max-size held-decs-max-size
```

13. (Optional) Specify the oldest age of outstanding Gate-Info requests.

```
[edit slot 0 jps rks-interface]
user@host# set held-decs-max-age held-decs-max-age
```

The maximum age must be greater than sent-message-timeout of the corresponding CMTS interface.

14. (Optional) Specify the size of billing correlation ID (BCID) cache.

```
[edit slot 0 jps rks-interface]
user@host# set bcid-cache-size bcid-cache-size
```

15. (Optional) Specify the oldest age of billing correlation ID (BCID) in cache.

```
[edit slot 0 jps rks-interface]
user@host# set bcid-cache-age bcid-cache-age
```

16. (Optional) Specify whether the default RKS pair is used when an application manager requests the use of an unconfigured RKS pair.

```
[edit slot 0 jps rks-interface]
user@host# set use-default-when-am-requests-unconfigured-rks
```

17. (Optional) Verify your configuration.

```
[edit slot 0 jps rks-interface]
user@host# show
```

Configuring RKS Pairs (SRC CLI)

By default, the JPS has four RKS pairs. All parameters that share the same RKS pair name configure the connection to that RKS pair. Any configured RKS pair can be used as the value for the default RKS pair or the RKS pair associated with a specific application manager.

NOTE: When running more than one JPS in a group to provide redundancy, all the JPSs in that group must have same RKS pair configuration (including the default RKS pair and any configured RKS pairs associated with a specific application manager).

Use the following configuration statement to configure the RKS pair:

```
slot number jps rks-interface rks-pair rks-pair-name {
    primary-address primary-address;
    primary-port primary-port;
    secondary-address secondary-address;
    secondary-port secondary-port;
}
```

To configure the RKS pair:

1. From configuration mode, access the configuration statement that configures the RKS pair. In this sample procedure, the RKS pair called pair1 is configured.

   user@host# **edit slot 0 jps rks-interface rks-pair pair1**

2. Specify the IP address of the primary RKS for this RKS pair.

   [edit slot 0 jps rks-interface rks-pair pair1]
   user@host# **set primary-address** *primary-address*

   If no value is specified, the RKS pair is not defined.

3. (Optional) Specify the UDP port on the primary RKS to which the JPS sends events.

   [edit slot 0 jps rks-interface rks-pair pair1]
   user@host# **set primary-port** *primary-port*

4. (Optional) Specify the IP address of the secondary RKS for this RKS pair.

   [edit slot 0 jps rks-interface rks-pair pair1]
   user@host# **set secondary-address** *secondary-address*

5. (Optional) Specify the UDP port on the secondary RKS to which the JPS sends events.

   [edit slot 0 jps rks-interface rks-pair pair1]
   user@host# **set secondary-port** *secondary-port*

6. (Optional) Verify your configuration.

```
[edit slot 0 jps rks-interface rks-pair pair1]
user@host# show
primary-address ;
primary-port 1813;
secondary-address ;
secondary-port 1813;
```

Related Topics

- Modifying the JPS Configuration (C-Web Interface)
- Configuring RKS Pairs (C-Web Interface)
- Specifying Connections to the Application Managers (SRC CLI) on page 86
- Viewing JPS RKS Statistics with the C-Web Interface on page 116

Configuring RKS Pairs for Associated Application Managers (SRC CLI)

By default, the JPS has four associated application managers. All parameters that share the same application manager name configure the RKS pair to which events associated with a specific application manager are sent.

Use the following configuration statement to configure the associated application manager:

slot *number* jps rks-interface am *am-name* {
  am-id *am-id*;

```
    rks-pair-name rks-pair-name;
    trusted;
}
```

To configure the associated application manager:

1. From configuration mode, access the configuration statement that configures the RKS pair for the associated application manager. In this sample procedure, the application manager name called 1 is configured.

   ```
   user@host# edit slot 0 jps rks-interface am 1
   ```

2. Specify the identifier of the application manager.

   ```
   [edit slot 0 jps rks-interface am 1]
   user@host# set am-id am-id
   ```

   If no value is specified, the RKS pair configuration is not defined for this application manager. If you must set trusted to true without defining the RKS pair configuration, you must specify a value for am-id and not specify a value for rks-pair-name.

3. (Optional) Specify the RKS pair that the JPS will send events to when those events are triggered by gate transitions associated with the application manager specified by am-id with the same application manager name (am-name).

   ```
   [edit slot 0 jps rks-interface am 1]
   user@host# set rks-pair rks-pair-name
   ```

   If no value is specified, the RKS pair configuration is not defined for this application manager. Use when you must set trusted to true without defining the RKS pair configuration.

4. (Optional) Specify whether this application manager is a trusted network element to the JPS.

   ```
   [edit slot 0 jps rks-interface am 1]
   user@host# set trusted
   ```

5. (Optional) Verify your configuration.

   ```
   [edit slot 0 jps rks-interface am 1]
   user@host# show
   ```

Specifying Connections to CMTS Devices (SRC CLI)

Use the following configuration statement to configure the policy server–to–CMTS interface (PKT-MM2) so that the policy server can communicate with CMTS devices:

```
slot number jps cmts-interface {
    cmts-addresses [cmts-addresses...];
    keepalive-interval keepalive-interval;
    synch-despite-unreachable-pep;
    synch-despite-pre-i03-pep;
    use-ssq-ssc-with-pre-i03-pep;
```

```
    local-address local-address;
    message-max-length message-max-length;
    message-read-buffer-size message-read-buffer-size;
    message-write-buffer-size message-write-buffer-size;
    open-connection-timeout open-connection-timeout;
    connection-open-retry-interval connection-open-retry-interval;
    sent-message-timeout sent-message-timeout;
    validate-pcmm-objects;
}
```

To configure the policy server–to–CMTS interface:

1. From configuration mode, access the configuration statement that configures the policy server–to–CMTS interface.

   ```
   user@host# edit slot 0 jps cmts-interface
   ```

2. Specify the IP addresses of all the CMTS devices to which the JPS will try to connect.

   ```
   [edit slot 0 jps cmts-interface]
   user@host# set cmts-addresses [cmts-addresses...]
   ```

3. (Optional) Specify the interval between keepalive messages sent from the COPS client (CMTS device) to the COPS server (the JPS). Changes apply only to COPS connections that are established after you make the change.

   ```
   [edit slot 0 jps cmts-interface]
   user@host# set keepalive-interval keepalive-interval
   ```

   A value of 0 means that no keepalive messages will be exchanged between the CMTS device and the JPS.

4. (Optional) Specify whether synchronization proceeds when the JPS receives a synchronization request from an application manager (such as the SAE) and the JPS is not connected to a CMTS device to which it should be connected.

   ```
   [edit slot 0 jps cmts-interface]
   user@host# set synch-despite-unreachable-pep
   ```

5. (Optional) Specify whether synchronization proceeds when the JPS receives a synchronization request from an application manager (such as the SAE) and the JPS is connected to a pre-PCMM I03 CMTS device.

   ```
   [edit slot 0 jps cmts-interface]
   user@host# set synch-despite-pre-i03-pep
   ```

6. (Optional) Specify whether synchronization includes both pre-PCMM I03 and PCMM I03 CMTS devices when the JPS receives a synchronization request from an application manager (such as the SAE) and the JPS is connected to a pre-PCMM I03 CMTS device. Relevant only when at least one pre-PCMM I03 CMTS device is connected and sync-despite-pre-i03-pep is specified as true.

   ```
   [edit slot 0 jps cmts-interface]
   user@host# set use-ssq-ssc-with-pre-i03-pep
   ```

7. (Optional) Specify the source IP address that the JPS uses to communicate with CMTS devices.

   ```
   [edit slot 0 jps cmts-interface]
   user@host# set local-address local-address
   ```

   If no value is specified and there is more than one local address, a random local address is used as the source address.

8. (Optional) Specify the maximum length of incoming messages.

   ```
   [edit slot 0 jps cmts-interface]
   user@host# set message-max-length message-max-length
   ```

9. (Optional) Specify the size of message read buffer.

   ```
   [edit slot 0 jps cmts-interface]
   user@host# set message-read-buffer-size message-read-buffer-size
   ```

10. (Optional) Specify the size of message write buffer.

    ```
    [edit slot 0 jps cmts-interface]
    user@host# set message-write-buffer-size message-write-buffer-size
    ```

11. (Optional) Specify the maximum time to wait for the initial PCMM messages to be exchanged after a TCP connection is established.

    ```
    [edit slot 0 jps cmts-interface]
    user@host# set open-connection-timeout open-connection-timeout
    ```

    The connection is dropped when initial PCMM messages are not exchanged within this time period.

12. (Optional) Specify the time to wait before the JPS tries to reconnect to CMTS devices.

    ```
    [edit slot 0 jps cmts-interface]
    user@host# set connection-open-retry-interval connection-open-retry-interval
    ```

13. (Optional) Specify the maximum time to wait for the sent messages to be exchanged after a TCP connection is established.

    ```
    [edit slot 0 jps cmts-interface]
    user@host# set sent-message-timeout sent-message-timeout
    ```

    This value must be less than the held-decs-max-age and pending-rks-event-max-age values for the corresponding RKS interface.

14. (Optional) Specify whether to validate PCMM objects received from PDPs.

    ```
    [edit slot 0 jps cmts-interface]
    user@host# set validate-pcmm-objects
    ```

15. (Optional) Verify your configuration.

    ```
    [edit slot 0 jps cmts-interface]
    user@host# show
    ```

```
cmts-addresses ;
keepalive-interval 60;
synch-despite-unreachable-pep;
synch-despite-pre-i03-pep;
local-address ;
message-max-length 204800;
message-read-buffer-size 1000000;
message-write-buffer-size 1000000;
open-connection-timeout 5;
connection-open-retry-interval 60;
sent-message-timeout 60;
validate-pcmm-objects;
```

Related Topics

- Specifying Connections to CMTS Devices (C-Web Interface)
- Specifying Connections to the Application Managers (SRC CLI) on page 86
- Viewing JPS State on page 106
- Viewing JPS CMTS Connections with the C-Web Interface on page 113

Modifying the Subscriber Configuration (SRC CLI)

To locate the CMTS device associated with a subscriber, the JPS maps the subscriber IP address in a message to the CMTS IP address to which the message must be delivered. This mapping specifies the subscriber IP pools associated with CMTS devices.

Use the following configuration statements to configure a CMTS device to which the JPS can connect and the pools of subscriber IP addresses that are managed by the CMTS device:

slot *number* jps cmts-registry cmts *cmts-ip* ...
slot *number* jps cmts-registry cmts *cmts-ip* range-pool *pool-index* {
  low *low*;
  high *high*;
}
slot *number* jps cmts-registry cmts *cmts-ip* subnet-pool *subnet* {
  exclude [*exclude*];
}

Tasks to modify subscriber configuration are:

1. “Configuring Subscriber IP Pools as IP Address Ranges (SRC CLI)” on page 96
2. “Configuring Subscriber IP Pools as IP Subnets (SRC CLI)” on page 96

Related Topics

- Modifying the Subscriber Configuration (C-Web Interface)
- Specifying Connections to CMTS Devices (SRC CLI) on page 92

Configuring Subscriber IP Pools as IP Address Ranges (SRC CLI)

To configure subscriber IP pools that are managed by the CMTS device as IP address ranges:

1. From configuration mode, access the configuration statement that configures the CMTS device to which the JPS can connect.

   user@host# **edit slot 0 jps cmts-registry cmts** *cmts-ip* **range-pool** *pool-index*

   Specify the IP address of the CMTS device and the address range pool index.

2. Specify the first IP address in the IP range for the pool of subscriber IP addresses that are managed by the CMTS device.

   [edit slot 0 jps cmts-registry cmts *cmts-ip* range-pool *pool-index*]
   user@host# **set low** *low*

3. Specify the last IP address in the IP range for the pool of subscriber IP addresses that are managed by the CMTS device.

   [edit slot 0 jps cmts-registry cmts *cmts-ip* range-pool *pool-index*]
   user@host# **set high** *high*

4. (Optional) Verify your configuration.

   ```
   [edit slot 0 jps cmts-registry]
   user@host# show
   ```

Related Topics

- Configuring Subscriber IP Pools as IP Address Ranges (C-Web Interface)
- Modifying the Subscriber Configuration (SRC CLI) on page 95
- Configuring Subscriber IP Pools as IP Subnets (SRC CLI) on page 96
- Specifying Connections to CMTS Devices (SRC CLI) on page 92

Configuring Subscriber IP Pools as IP Subnets (SRC CLI)

To configure subscriber IP pools that are managed by the CMTS device as IP subnets:

1. From configuration mode, access the configuration statement that configures the CMTS device to which the JPS can connect.

   user@host# **edit slot 0 jps cmts-registry cmts** *cmts-ip* **subnet-pool** *subnet*

   Specify the IP address of the CMTS device and the IP address and mask of the subnet for the pool of subscriber IP addresses.

2. (Optional) Specify the IP addresses of the subnet that are excluded from the subscriber IP pool managed by the CMTS device.

   [edit slot 0 jps cmts-registry cmts *cmts-ip* subnet-pool *subnet*]

```
user@host# set exclude [exclude...]
```

3. (Optional) Verify your configuration.

```
[edit slot 0 jps cmts-registry]
user@host# show
```

Related Topics

- Configuring Subscriber IP Pools as IP Subnets (C-Web Interface)
- Modifying the Subscriber Configuration (SRC CLI) on page 95
- Configuring Subscriber IP Pools as IP Address Ranges (SRC CLI) on page 96
- Specifying Connections to CMTS Devices (SRC CLI) on page 92

Configuring the SAE to Interact with the JPS (SRC CLI)

You must configure the SAE as an application manager to allow it to interact with PCMM-compliant policy servers. The policy server acts as a policy decision point that manages the relationships between application managers and CMTS devices. Policy servers that manage the same group of CMTS devices are grouped together and are simultaneously active. The policy server group provides a way for the SAE to communicate with any CMTS device that is managed by a policy server in the policy server group. To provide redundancy, the SAEs are grouped in an SAE community that connects to a policy server group. Only one of the SAEs in the SAE community is active. The active SAE establishes connections to all the policy servers in the policy server group. The active SAE will fail over to a redundant SAE only when it loses the connection to all the policy servers in the policy server group. State synchronization enables the SAE to synchronize its state with all the CMTS devices connected to a policy server group.

The tasks to configure the SAE as an application manager are:

- "Specifying Application Managers for the Policy Server (SRC CLI)" on page 98
- "Specifying Application Manager Identifiers for Policy Servers (SRC CLI)" on page 99
- "Adding Objects for Policy Servers to the Directory (SRC CLI)" on page 100
- "Configuring Initialization Scripts (SRC CLI)" on page 101
- "Enabling State Synchronization (SRC CLI)" on page 101

Related Topics

- Overview of the JPS on page 75
- Configuring the SAE to Interact with the JPS (C-Web Interface)
- Configuring the JPS (SRC CLI) on page 81
- Initially Configuring the SAE

Specifying Application Managers for the Policy Server (SRC CLI)

To specify the SAE community that connects to a policy server group, you need to add an application manager group object to the directory.

Use the following configuration statements to specify the application manager for the policy server:

```
shared network application-manager-group name {
    description description;
    application-manager-id application-manager-id;
    connected-sae [connected-sae...];
    pdp-group pdp-group;
    local-address-pools [local-address-pools...];
    managing-sae-ior managing-sae-ior;
}
```

To add an application manager group:

1. From configuration mode, access the configuration statement that specifies the application managers.

   user@host# **edit shared network application-manager-group** *name*

2. (Optional) Specify information about the SAE community.

   [edit shared network application-manager-group *name*]
   user@host# **set description** *description*

3. (Optional) Specify the unique identifier within the domain of the service provider for the application manager that handles the service session (Application Manager Tag) as a 2-byte unsigned integer.

   [edit shared network application-manager-group *name*]
   user@host# **set application-manager-id** *application-manager-id*

4. (Optional) Specify the SAEs that are connected to the specified policy server group. This list becomes the community of SAEs.

   [edit shared network application-manager-group *name*]
   user@host# **set connected-sae** [*connected-sae*...]

   When you modify a community, wait for passive session stores of the new community members to be updated before you shut down the current active SAE. Otherwise, a failover from the current active SAE to the new member is triggered immediately, and the new member's session store may not have received all data from the active SAE's session store.

5. (Optional) Specify the name of the policy server group associated with this SAE community.

   [edit shared network application-manager-group *name*]
   user@host# **set pdp-group** *pdp-group*

6. (Optional) Specify the list of IP address pools that the specified PDP group currently manages and stores.

   ```
   [edit shared network application-manager-group name]
   user@host# set local-address-pools local-address-pools
   ```

   You must configure a local address pool if you are using the NIC so that the NIC can resolve the IP-to-SAE mapping. See “Using the NIC Resolver” on page 102.

7. (Optional) Specify the Common Object Request Broker Architecture (CORBA) reference for the SAE managing this policy server group.

   ```
   [edit shared network application-manager-group name]
   user@host# set managing-sae-ior managing-sae-ior
   ```

   The **amIorPublisher** script provides this information when the SAE connects to the policy server. If you do not select this script when configuring initialization scripts, enter a value. For information about configuring initialization scripts, see “Configuring Initialization Scripts (SRC CLI)” on page 101.

Related Topics

- JPS Interfaces on page 76
- Specifying Application Managers for the Policy Server (C-Web Interface)
- Specifying Application Manager Identifiers for Policy Servers (SRC CLI) on page 99
- Viewing JPS State on page 106
- Viewing JPS AM Statistics with the C-Web Interface on page 110

Specifying Application Manager Identifiers for Policy Servers (SRC CLI)

The application manager identifier (AMID) identifies the application manager (such as the SAE) in messages sent to and from the policy server. The SAE constructs the AMID value by concatenating two fields: Application Manager Tag and Application Type.

The Application Manager Tag value is obtained from the specification of application managers for policy servers. See “Specifying Application Managers for the Policy Server (SRC CLI)” on page 98.

The Application Type value is obtained during service activation from the specification of the PCMM Application Type value when you configure normal services.

For more information about configuring services, see Adding a Normal Service (SRC CLI).

Related Topics

- JPS Interfaces on page 76
- Specifying Application Manager Identifiers for Policy Servers (C-Web Interface)
- Specifying Policy Server Identifiers in Messages (SRC CLI) on page 83
- Adding Objects for Policy Servers to the Directory (SRC CLI) on page 100
- Viewing JPS State on page 106

Adding Objects for Policy Servers to the Directory (SRC CLI)

To communicate with policy servers, the SAE creates and manages pseudointerfaces that it associates with a policy decision point object in the directory. Each policy server in the SRC network must appear in the directory as a policy decision point object.

Use the following configuration statements to specify the policy server as a policy decision point:

```
shared network policy-decision-point name {
    description description;
    pdp-address pdp-address;
    pdp-group pdp-group;
}
```

To add a policy server to the directory with the SRC CLI:

1. From configuration mode, access the configuration statement that configures the policy decision point.

   ```
   user@host# edit shared network policy-decision-point name
   ```

2. (Optional) Specify information about the policy server.

   ```
   [edit shared network policy-decision-point name]
   user@host# set description description
   ```

3. (Optional) Specify the IP address of the policy server. The SAE uses this address to establish a COPS connection with the policy server.

   ```
   [edit shared network policy-decision-point name]
   user@host# set pdp-address pdp-address
   ```

4. (Optional) Specify the name of the policy server group.

   ```
   [edit shared network policy-decision-point name]
   user@host# set pdp-group pdp-group
   ```

5. Create an SAE community for the policy servers. See “Specifying Application Managers for the Policy Server (SRC CLI)” on page 98.

Related Topics

- Policy Components
- Adding Objects for Policy Servers to the Directory (C-Web Interface)
- Configuring Initialization Scripts (SRC CLI) on page 101
- Specifying Application Manager Identifiers for Policy Servers (SRC CLI) on page 99
- Enabling State Synchronization (SRC CLI) on page 101

Configuring Initialization Scripts (SRC CLI)

When the SAE establishes a connection with a policy server, it runs an initialization script to customize the setup of the connection.

Use the following configuration statement to configure the initialization script:

```
shared sae configuration driver scripts {
  pcmm pcmm;
}
```

To configure initialization scripts for the SAE:

1. From configuration mode, access the configuration statement that configures the initialization scripts.

   ```
   user@host# edit shared sae configuration driver scripts
   ```

2. Specify the initialization script for a PCMM environment.

   ```
   [edit shared sae configuration driver scripts]
   user@host# set pcmm pcmm
   ```

   The script is run when the connection between a policy server and the SAE is established and again when the connection is dropped. For the JPS, we recommend setting this value to amIorPublisher.

Related Topics

- Configuring Initialization Scripts (C-Web Interface)
- Specifying Application Managers for the Policy Server (SRC CLI) on page 98
- Configuring the SAE to Interact with the JPS (SRC CLI) on page 97
- Specifying Initialization Scripts on the SAE (SRC CLI)
- Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers

Enabling State Synchronization (SRC CLI)

State synchronization is achieved when the SAE is required to communicate with the policy server over the COPS connection.

Use the following configuration statement to configure state synchronization:

```
shared sae configuration driver pcmm {
  disable-full-sync;
  disable-pcmm-i03-policy;
  session-recovery-retry-interval session-recovery-retry-interval;
}
```

To enable state synchronization with policy servers:

1. From configuration mode, access the configuration statement that configures the PCMM device driver.

   user@host# edit shared sae configuration driver pcmm

2. Specify whether state synchronization with the PCMM policy servers is disabled.

   [edit shared sae configuration driver pcmm]
   user@host# set disable-full-sync

   When using other PCMM-compliant policy servers (instead of the JPS), we recommend setting this value to true.

3. Specify whether PCMM I03 policies are disabled when the SAE is deployed with pre-PCMM I03 CMTS devices.

   [edit shared sae configuration driver pcmm]
   user@host# set disable-pcmm-i03-policy

   When there are pre-PCMM I03 CMTS devices in the network, you must set this value to true.

4. Specify the time interval between attempts by the SAE to restore service sessions that are still being recovered in the background when state synchronization completes with a state-data-incomplete error.

   [edit shared sae configuration driver pcmm]
   user@host# set session-recovery-retry-interval *session-recovery-retry-interval*

   We recommend setting this value to 3600000 (1 hour) or longer.

Related Topics

- Enabling State Synchronization (C-Web Interface)
- Configuring the SAE to Interact with the JPS (SRC CLI) on page 97
- Viewing JPS State on page 106

Using the NIC Resolver

If you are using the NIC to map the subscriber IP address to the SAE, you need to configure a NIC host. The NIC system uses IP address pools to map IP addresses to SAEs. You configure the local address pools in the application manager configuration for a policy server group. These pools are published in the NIC. The NIC maps subscriber IP addresses in requests received through the portal or Advanced Services Gateway to the policy server group that currently manages that CMTS device. For information about configuring the SAE for policy servers, see “Specifying Application Managers for the Policy Server (SRC CLI)” on page 98 .

The OnePopPcmm sample configuration data supports this scenario for a PCMM environment in which you use the assigned IP subscriber method to log in subscribers and in which you use the NIC to determine the subscriber’s SAE. The OnePopPcmm configuration supports one point of presence (POP). NIC replication can be used to

provide high availability. The realm for this configuration accommodates the situation in which IP pools are configured locally on each application manager group object.

The resolution process takes a subscriber's IP address as the key and returns a reference to the SAE managing this subscriber as the value.

The following agents collect information for resolvers in this realm:

- Directory agent PoolVr collects and publishes information about the mappings of IP pools to the policy server group.
- Directory agent VrSaeId collects and publishes information about the mappings of policy server groups to SAEs.

For more information about configuring the NIC, see Configuring the NIC (SRC CLI).

Related Topics

- Overview of a PCMM Environment on page 41
- OnePopPcmm Scenario

Managing the JPS

After you have installed the JPS and applied the local configuration of the JPS, you can perform these tasks:

- Starting the JPS (SRC CLI) on page 103
- Restarting the JPS (SRC CLI) on page 103
- Stopping the JPS (SRC CLI) on page 104
- Displaying JPS Status (SRC CLI) on page 104

Starting the JPS (SRC CLI)

You must start the JPS when you install the JPS without rebooting the JPS host.

To start the JPS:

```
user@host> enable component jps
```

The system responds with a start message. If the JPS is already running, the system responds with a warning message.

Related Topics

- Starting the JPS (C-Web Interface)
- Stopping the JPS (SRC CLI) on page 104
- Configuring the JPS (SRC CLI) on page 81
- Monitoring the JPS on page 105
- Overview of the JPS on page 75

Restarting the JPS (SRC CLI)

To restart the JPS:

```
user@host> restart component jps
```

The system responds with a start message. If the JPS is already running, the system responds with a shutdown message and then a start message.

Related Topics

- Restarting the JPS (C-Web Interface)
- Stopping the JPS (SRC CLI) on page 104
- Configuring the JPS (SRC CLI) on page 81
- Monitoring the JPS on page 105
- Overview of the JPS on page 75

Stopping the JPS (SRC CLI)

To stop the JPS:

```
user@host> disable component jps
```

The system responds with a shutdown message. If the JPS is not running when you issue the command, the system responds with the command prompt.

To start the JPS, see “Starting the JPS (SRC CLI)” on page 103.

Related Topics

- Stopping the JPS (C-Web Interface)
- Restarting the JPS (SRC CLI) on page 103
- Monitoring the JPS on page 105
- Overview of the JPS on page 75

Displaying JPS Status (SRC CLI)

Purpose Display the JPS status.

Action

```
user@host> show component
```

The system responds with a status message.

Related Topics

- Displaying JPS Status (C-Web Interface)
- Monitoring the JPS on page 105
- Viewing JPS State on page 106

CHAPTER 11

Monitoring the JPS (SRC CLI)

Monitoring the JPS

Purpose Monitor the following JPS information:

- The basic health indicators for the server process
- The current state of the JPS, such as the current network connections or recent performance statistics

Action

```
user@host> show jps statistics
```

Related Topics

- Displaying JPS Status (SRC CLI) on page 104
- Viewing JPS State on page 106
- Viewing Server Process Information on page 105
- Configuring the JPS (SRC CLI) on page 81
- Overview of the JPS on page 75

Viewing Server Process Information

Purpose View information about the server process.

Action

```
user@host> show jps statistics process
```

Related Topics

- Viewing Information About the JPS Server Process with the C-Web Interface on page 109
- Monitoring the JPS on page 105
- Viewing JPS State on page 106
- Overview of the JPS on page 75

Viewing JPS State

You can monitor the current state of the JPS by:

Viewing Performance Statistics for the JPS Interfaces

Purpose View performance statistics for JPS interfaces.

Action To view recent performance statistics for the application manager-to-policy server interface:

```
user@host> show jps statistics am
```

To view recent performance statistics for the policy server-to-CMTS interface:

```
user@host> show jps statistics cmts
```

To view recent performance statistics for the policy server-to-RKS interface:

```
user@host> show jps statistics rks
```

Viewing Network Connections for the Application Manager

Purpose View network connections for the application manager.

Action To view information about the current JPS network connections for all the application managers:

```
user@host> show jps statistics am connections
```

To view information about the current JPS network connections for a specific application manager:

```
user@host> show jps statistics am connections  ip-address ip-address
```

Enter all or part of the IP address to list connections for all matching addresses.

Viewing Network Connections for the CMTS Device

Purpose View network connections for the CMTS Device.

Action To view information about the current JPS connections for all the CMTS devices:

```
user@host> show jps statistics cmts connections
```

To view information about the current JPS connections for a specific CMTS device:

```
user@host> show jps statistics cmts connections ip-address ip-address
```

Enter all or part of the IP address to list connections for all matching addresses.

Viewing Performance Statistics for the CMTS Locator

Purpose View information about the recent performance statistics for the CMTS locator.

Action

```
user@host> show jps statistics cmts-locator
```

Viewing Message Handler Information

Purpose View message handler information.

Action To view information about the JPS message handler and message flows:

```
user@host> show jps statistics message-handler
user@host> show jps statistics message-handler message-flow
```

To view information about specific JPS message flows:

```
user@host> show jps statistics message-handler message-flow  id id
```

Enter all or part of the message flow identifier to list all matching message flows.

Related Topics

- Displaying JPS Status (SRC CLI) on page 104
- Monitoring the JPS on page 105
- Viewing Server Process Information on page 105
- Overview of the JPS on page 75

CHAPTER 12

Monitoring the JPS with the C-Web Interface

Viewing Information About the JPS Server Process with the C-Web Interface

Purpose View information about the JPS server process.

Action Click **JPS** >**Statistics**>**Process**.

The Statistics/Process pane displays the JPS server process information.

Related Topics

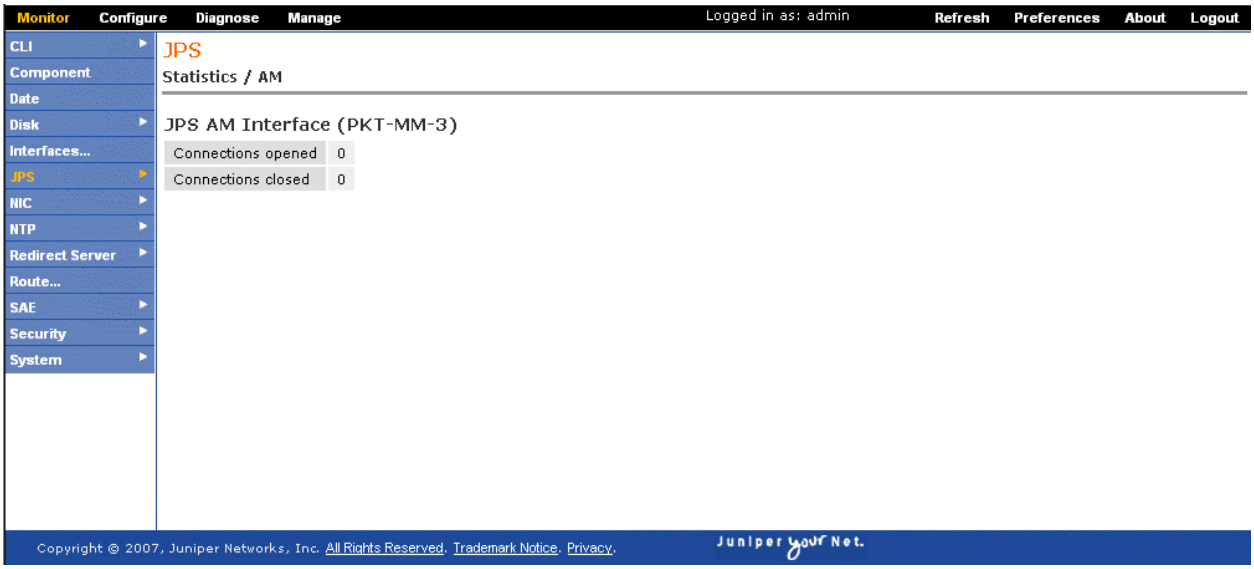
- Viewing Server Process Information on page 105
- Configuring the JPS (C-Web Interface)
- Overview of the JPS on page 75

Viewing JPS AM Statistics with the C-Web Interface

Purpose View information about recent performance statistics for the application manager–to–policy server interface.

Action Click **JPS**>**Statistics**>**AM**.

The Statistics/AM pane displays performance statistics for the application manager–to–policy server interface.



Related Topics

- Viewing JPS State on page 106
- Modifying the JPS Configuration (C-Web Interface)
- Viewing JPS AM Connections with the C-Web Interface on page 111
- Overview of the JPS on page 75

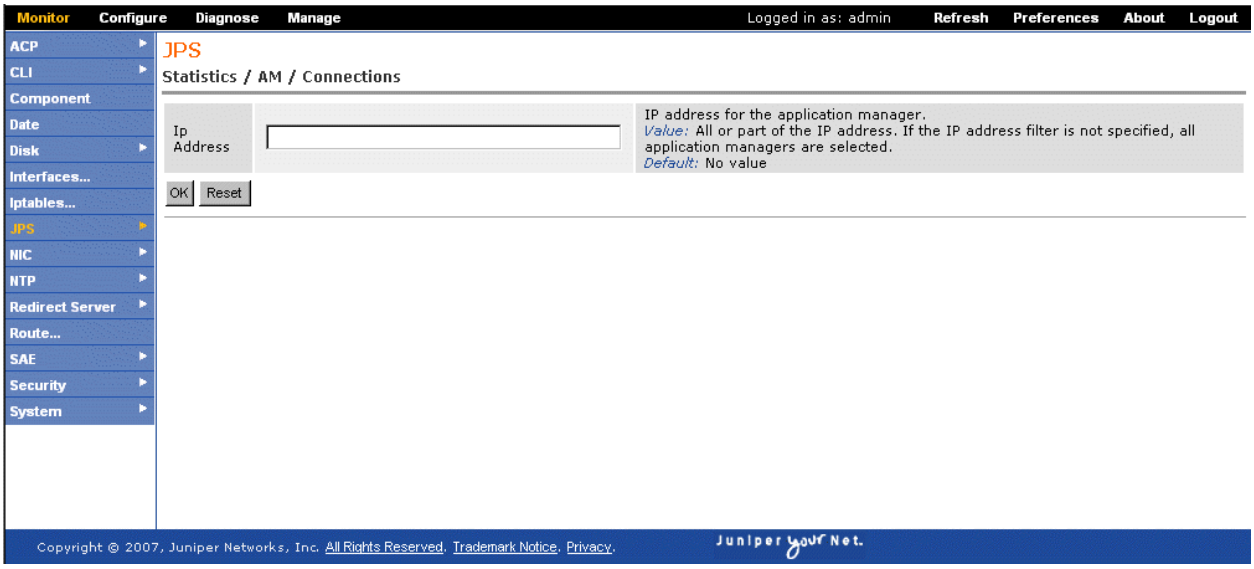
Viewing JPS AM Connections with the C-Web Interface

Purpose View information about the current JPS network connections for the application manager.

Action

1. Click **JPS**>**Statistics**>**AM**>**Connections**.

   The Statistics/AM/Connections pane appears.



2. In the IP Address box, enter the IP address, or leave the box blank to display all AM connections.
3. Click **OK**.

   The Statistics/AM/Connections pane displays the AM connection statistics.

Related Topics

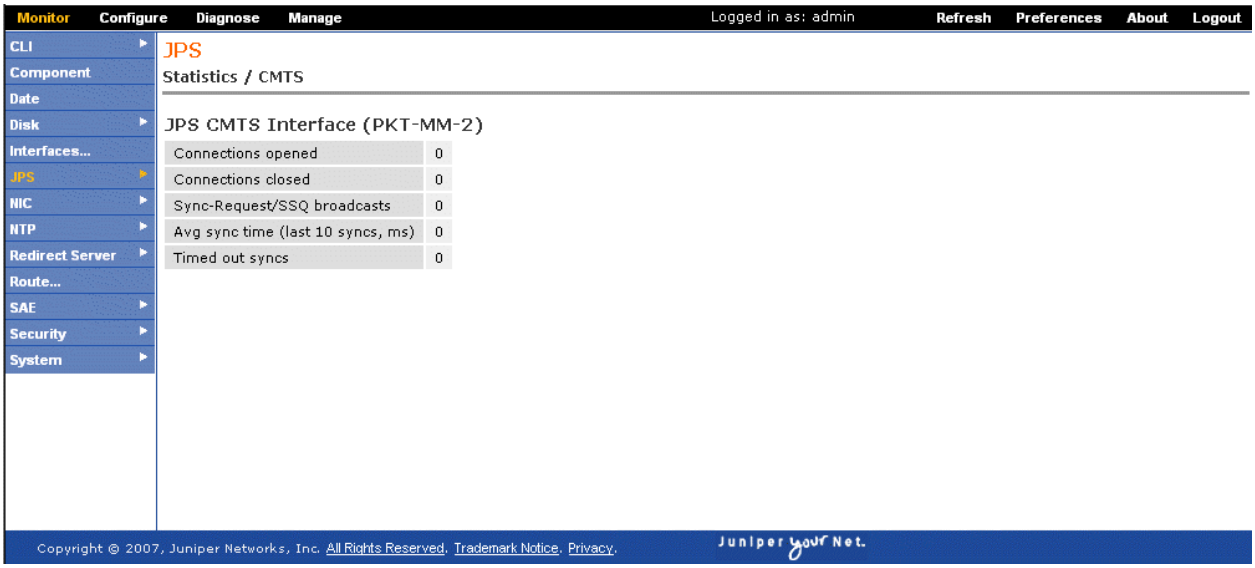
- Viewing JPS State on page 106
- Modifying the JPS Configuration (C-Web Interface)
- Viewing JPS AM Statistics with the C-Web Interface on page 110
- Overview of the JPS on page 75

Viewing JPS CMTS Statistics with the C-Web Interface

Purpose View information about recent performance statistics for the policy server–to–CMTS interface.

Action Click **JPS**>**Statistics**>**CMTS**.

The Statistics/CMTS pane displays statistics for the policy server–to–CMTS interface.



Related Topics

- Viewing JPS State on page 106
- Specifying Connections to CMTS Devices (C-Web Interface)
- Viewing JPS CMTS Connections with the C-Web Interface on page 113
- Viewing JPS CMTS Locator Statistics with the C-Web Interface on page 113
- Overview of the JPS on page 75

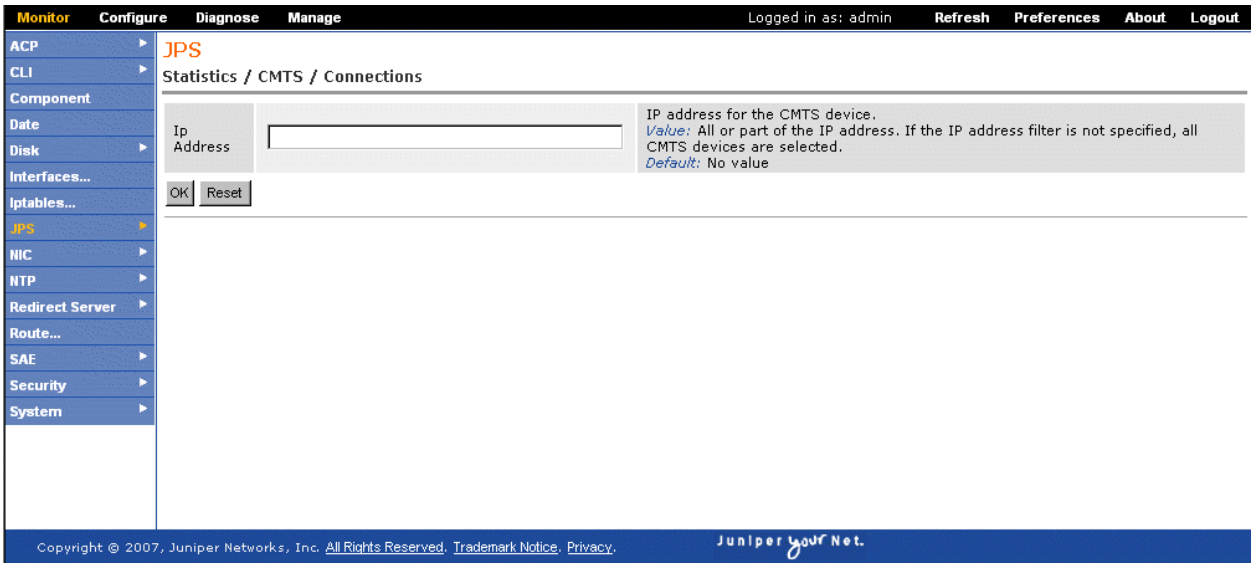
Viewing JPS CMTS Connections with the C-Web Interface

Purpose View information about the current JPS network connections for the CMTS device.

Action

1. Click **JPS**>**Statistics**>**CMTS**>**Connections**.

   The Statistics/CMTS/Connections pane appears.



2. In the IP Address box, enter the IP address, or leave the box blank to display all CMTS connections.
3. Click **OK**.

   The Statistics/CMTS/Connections pane displays the CMTS connection statistics.

Related Topics

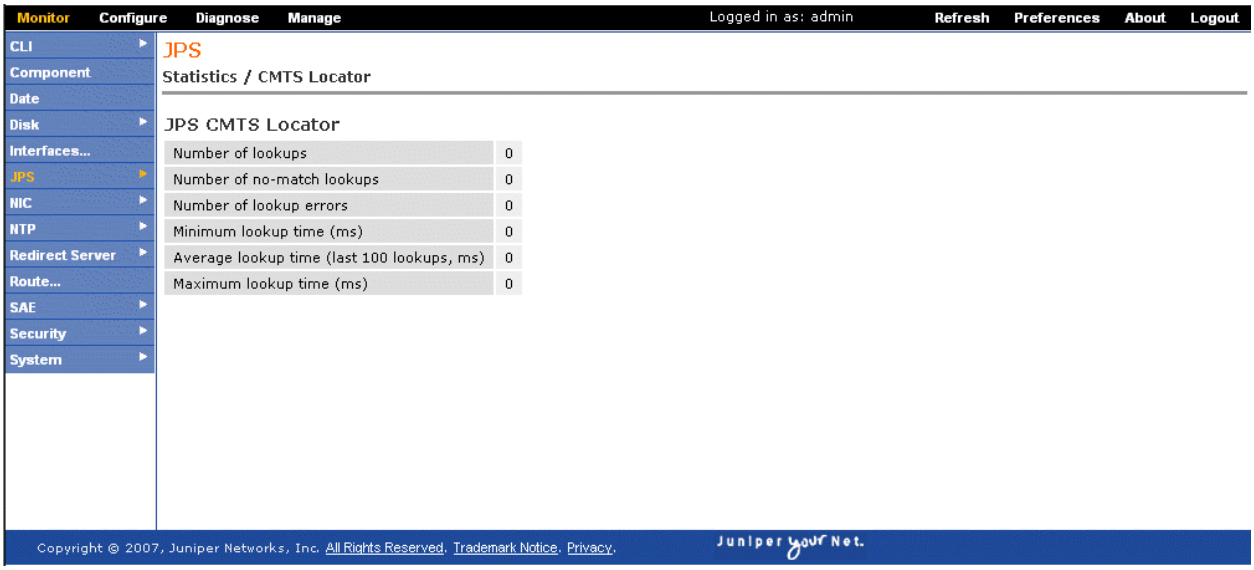
- Viewing JPS State on page 106
- Specifying Connections to CMTS Devices (C-Web Interface)
- Viewing JPS CMTS Statistics with the C-Web Interface on page 112
- Viewing JPS CMTS Locator Statistics with the C-Web Interface on page 113
- Overview of the JPS on page 75

Viewing JPS CMTS Locator Statistics with the C-Web Interface

Purpose View information about the recent performance statistics for the CMTS locator.

Action Click **JPS**>**Statistics**>**CMTS Locator**.

The Statistics/CMTS Locator pane displays the CMTS locator statistics.



Related Topics

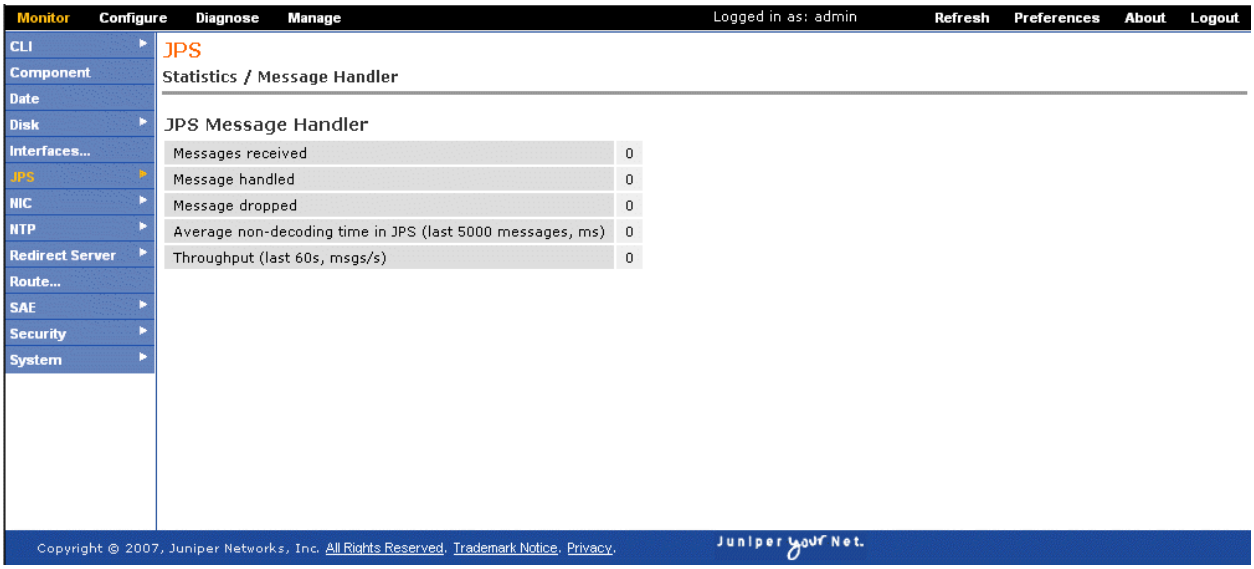
- Viewing JPS State on page 106
- Specifying Connections to CMTS Devices (C-Web Interface)
- Viewing JPS CMTS Statistics with the C-Web Interface on page 112
- Viewing JPS CMTS Connections with the C-Web Interface on page 113
- Overview of the JPS on page 75

Viewing JPS Message Handler Statistics with the C-Web Interface

Purpose View information about the JPS message handler.

Action Click **JPS**>**Statistics**>**Message Handler**.

The Statistics/Message Handler pane displays the JPS message handler statistics.



Related Topics

- Viewing JPS State on page 106
- Viewing JPS Message Flow Statistics with the C-Web Interface on page 115
- Overview of the JPS on page 75

Viewing JPS Message Flow Statistics with the C-Web Interface

Purpose View information about JPS message flows.

Action

1. Click **JPS**>**Statistics**>**Message Handler**>**Message Flows**.

   The Statistics/Message Handler/Message Flow pane appears.

2. In the ID box, enter a message flow ID, or leave the box blank to display statistics for all message flows.
3. Click **OK**.

   The Statistics/Message Handler/Message Flow pane displays the message flow statistics.

Related Topics

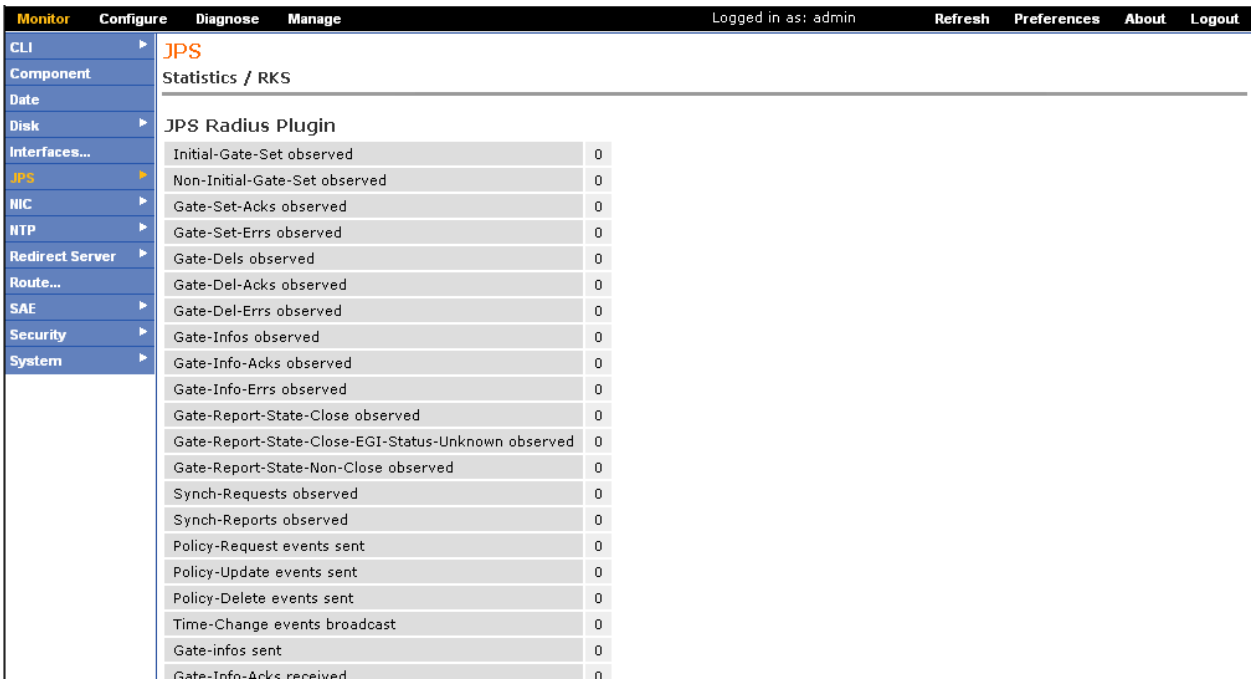
- Viewing JPS State on page 106
- Viewing JPS Message Handler Statistics with the C-Web Interface on page 114
- Overview of the JPS on page 75

Viewing JPS RKS Statistics with the C-Web Interface

Purpose View recent performance statistics for the policy server–to–record keeping server (RKS) interface.

Action Click **JPS**>**Statistics**>**RKS**.

The Statistics/RKS pane displays statistics for the policy server–to–RKS interface.



Related Topics

- Viewing JPS AM Statistics with the C-Web Interface on page 110
- Viewing JPS CMTS Statistics with the C-Web Interface on page 112
- Configuring RKS Pairs (C-Web Interface)
- Modifying the JPS Configuration (C-Web Interface)

- Overview of the JPS on page 75

PART 3

Managing Services on RADIUS Devices

CHAPTER 13

Managing Services on Third-Party Devices in the SRC Network

Overview of CoA Script Service

The 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 Topics

- Configuring CoA Script Services on page 122
- Configuring Subscriptions to the CoA Script Service on page 126
- Configuring Monitoring Agent to Receive RADIUS Accounting Messages on page 122
- Parameters for Sample CoA Script Service on page 125
- Example: Using the Sample CoA Script Service on page 126

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 122
- "Creating the CoA Script Service (SRC CLI)" on page 123
- "Configuring the CoA Script Service (SRC CLI)" on page 124
- "Configuring Subscriptions to the CoA Script Service" on page 126

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 126 .

Related Topics

- Overview of CoA Script Service on page 121
- Defining RADIUS Attributes for CoA Requests with the API on page 127
- Setting Up Script Services
- Parameters for Sample CoA Script Service on page 125

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 Topics

- Configuring the CoA Script Service (SRC CLI) on page 124
- Defining RADIUS Attributes for CoA Requests with the API on page 127

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 Overview of SRC Script Services.

Related Topics

- Overview of CoA Script Service on page 121
- Configuring Subscriptions to the CoA Script Service on page 126
- Configuring CoA Script Services on page 122
- Configuring the CoA Script Service (SRC CLI) on page 124
- Parameters for Sample CoA Script Service on page 125

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 125.

Related Topics

- Overview of CoA Script Service on page 121
- Configuring Subscriptions to the CoA Script Service on page 126
- Creating the CoA Script Service (SRC CLI) on page 123
- Configuring CoA Script Services on page 122
- Example: Using the Sample CoA Script Service on page 126

Parameters for Sample CoA Script Service

Table 7 on page 125 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 7: 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.
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 127).

Related Topics

- Overview of CoA Script Service on page 121
- Configuring Monitoring Agent to Receive RADIUS Accounting Messages on page 122
- Creating the CoA Script Service (SRC CLI) on page 123
- Configuring CoA Script Services on page 122
- Example: Using the Sample CoA Script Service on page 126

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 Topics

- Overview of CoA Script Service on page 121
- Configuring CoA Script Services on page 122
- Configuring the CoA Script Service (SRC CLI) on page 124
- Example: Using the Sample CoA Script Service on page 126

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 124. For information about passing the values through

the SAE core API, see “Defining RADIUS Attributes for CoA Requests with the API” on page 127.

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

   For more information about subscriptions, see Overview of Subscriptions.

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 Topics

- Overview of CoA Script Service on page 121
- Configuring Subscriptions to the CoA Script Service on page 126
- Configuring CoA Script Services on page 122
- Creating the CoA Script Service (SRC CLI) on page 123

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 124)
- 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.gzfile:

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

Related Topics

- Overview of CoA Script Service on page 121
- Configuring CoA Script Services on page 122
- Creating the CoA Script Service (SRC CLI) on page 123
- Configuring Monitoring Agent to Receive RADIUS Accounting Messages on page 122

CHAPTER 14

Managing Services on RADIUS-Enabled Devices

Overview of the IMS AAA Server Integration

When the Juniper IMS AAA Server is integrated in the SRC network, the SRC software can use the IMS AAA Server to dynamically manage services on RADIUS-enabled devices. The RADIUS capabilities of the IMS AAA Server allow the SRC software to be aware of the subscriber activity and to make dynamic RADIUS requests using these RADIUS features:

- Authentication, authorization, and accounting (AAA)
- Change-of-authorization (CoA) message
- Disconnect message (DM)

The SRC software communicates with the IMS AAA Server using the DIAMETER protocol. The IMS AAA Server uses RADIUS AAA messages to communicate with the RADIUS server and the Network Access Server (NAS). The IMS AAA Server acts as a proxy to convert Diameter messages to RADIUS messages and vice versa. The IMS AAA Server also performs conversion between Diameter attribute value pairs (AVPs) and RADIUS attributes.

When integrated in the SRC network, the IMS AAA Server can provide:

- Device abstraction and shared secrets for the NAS device
- Accounting support for subscriber sessions and service sessions

- CoA and DM support
- Service parameter changes

Managing Dynamic Services

You can integrate the IMS AAA Server to support management of services on RADIUS-enabled devices in an SRC network. The IMS AAA Server will act as a proxy to intercept messages between the NAS device and the RADIUS server so that the SRC software does not need to know device details. You can configure the services, policies, and parameters with the SRC software independent of the NAS device. The SRC software communicates with the IMS AAA Server using Diameter messages to dynamically manage services for a subscriber session. The IMS AAA Server converts the Diameter messages to RADIUS messages and makes dynamic RADIUS requests to the NAS device.

The SRC software includes a Diameter server that forwards AAR, ACR, and STR messages from the IMS AAA Server to the AAA device driver in the SAE and that forwards PPR and ASR messages from the AAA device driver to the IMS AAA Server. These Diameter messages perform these functions:

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

You configure NAS groups and a AAA device driver for each NAS group hosted by the SAE. You also configure the services, policies, and parameters that the IMS AAA Server will use for service activation on the NAS device. You will need to provide specific information for the custom router template on the IMS AAA Server.

The custom router template (deviceModels.xml) lists the parameters needed for service activation on a NAS device (controlledDeviceModel element in the template). The IMS AAA Server 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 IMS AAA Server.
- Configure the Diameter application.
- Configure the NAS groups.
- Configure the SAE to manage AAA devices.
- Configure AAA policies.

Configuring the IMS AAA Server

Tasks to set up the IMS AAA Server in the SRC network are:

- Configure the local elements for the IMS AAA Server.
  - Identification—IMS AAA Server
  - Diameter configuration—Transport port protocol
  - RADIUS configuration—AAA ports
- Configure the remote network elements.
  - The Diameter elements are for the C Series Controller and include: Origin host, IP address, port protocol, and function (SRC).
  - The RADIUS elements are for the NAS device and include: Name, IP address, and function.
- Review the custom router template for the NAS device. The service template element lists the parameters needed for service activation on the NAS device.

For information about configuring the IMS AAA Server with the IMS AAA Server Administrator, see the *IMS AAA Server Administration Guide*.

Configuring the Diameter Application (SRC CLI)

Tasks to configure the Diameter application are:

- Configuring the Diameter Application Properties on page 131
- Configuring the Diameter Client Properties on page 134
- Configuring the Diameter Server Properties on page 135
- Configuring Logging Destinations on page 135

Configuring the Diameter Application Properties

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;
   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 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) 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.

11. (Optional) Specify whether the peer connection is in debug mode.

```
[edit system diameter]
user@host# set debug-mode
```

12. (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)
```

13. (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.

14. (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.

15. (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.

16. (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).

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 a AAA peer group, the SAEs on the listed hosts will 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 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.

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.

Selecting 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 IMS AAA 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. Only applicable if the transaction variable is request-packet.

   [edit shared network nas-group *name* routes *name* term *name*]
   user@host# set dictionary-attribute (user-name | user-password | chap-password | nas-ip-address | nas-port | service-type | framed-protocol | framed-ip-address | framed-ip-netmask | framed-mtu | framed-compression | login-ip-host | callback-number | state | vendor-specific | called-station-id | calling-station-id | nas-identifier | login-lat-service | login-lat-node | login-lat-group | chap-challenge | nas-port-type | port-limit | login-lat-port)

6. Specify the operator for criterion matching.

   [edit shared network nas-group *name* routes *name* term *name*]
   user@host# set operator (equals | not_equal | present | not_present | prefix | suffix | range)

7. (Optional) Specify the value to be matched by the target.

   [edit shared network nas-group *name* routes *name* term *name*]
   user@host# set value *value*

8. (Optional) Specify the low end of the range criterion.

   [edit shared network nas-group *name* routes *name* term *name*]
   user@host# set low *low*

9. (Optional) Specify the high end of the range criterion.

   [edit shared network nas-group *name* routes *name* term *name*]
   user@host# set high *high*

Configuring the SAE to Manage AAA Devices

Use the following configuration statements to configure the AAA device driver:

```
shared sae configuration driver aaa {
    sae-community-manager sae-community-manager;
    origin-host origin-host;
    origin-realm origin-realm;
    keep-alive-timeout keep-alive-timeout;
    registry-retry-interval registry-retry-interval;
    reply-timeout reply-timeout;
    sequential-message-timeout sequential-message-timeout;
    transient-session-timeout transient-session-timeout;
    max-update-interval max-update-interval;
    update-grace-period update-grace-period;
    resume-unrecovered;
    thread-pool-size thread-pool-size;
    thread-idle-timeout thread-idle-timeout;
}
```

To configure the AAA device driver:

1. From configuration mode, access the configuration statements for the AAA device driver.

   user@host# edit shared sae configuration driver aaa

2. Specify the name of the community manager.

   [edit shared sae configuration driver aaa]
   user@host# set sae-community-manager *sae-community-manager*

3. (Optional) Specify the fully qualified domain name used to identify this host.

   [edit shared sae configuration driver aaa]
   user@host# set origin-host *origin-host*

4. (Optional) Specify the DNS name of the machine used to identify this host.

   [edit shared sae configuration driver aaa]
   user@host# set origin-realm *origin-realm*

5. (Optional) Specify the keepalive timeout before the registry to a Diameter server expires.

   [edit shared sae configuration driver aaa]
   user@host# set keep-alive-timeout *keep-alive-timeout*

6. (Optional) Specify the interval between retrying a failed registry to a Diameter server.

   [edit shared sae configuration driver aaa]
   user@host# set registry-retry-interval *registry-retry-interval*

7. (Optional) Specify the timeout before a request sent to a Diameter server expires.

   [edit shared sae configuration driver aaa]
   user@host# set reply-timeout *reply-timeout*

8. (Optional) Specify the timeout before an expected message expires.

   [edit shared sae configuration driver aaa]
   user@host# set sequential-message-timeout *sequential-message-timeout*

9. (Optional) Specify the timeout before a temporary session expires.

   [edit shared sae configuration driver aaa]
   user@host# set transient-session-timeout *transient-session-timeout*

10. (Optional) Specify the maximum interval between interim updates for a subscriber session.

    [edit shared sae configuration driver aaa]
    user@host# set max-update-interval *max-update-interval*

11. (Optional) Specify the grace period in which to expect an interim update for a subscriber session.

    [edit shared sae configuration driver aaa]
    user@host# set update-grace-period *update-grace-period*

12. (Optional) Specify whether to resume a subscriber session that has failed to recover from a failover.


```
[edit shared sae configuration driver aaa]
user@host# set resume-unrecovered
```

13. (Optional) Specify the number of working threads that process requests.

```
[edit shared sae configuration driver aaa]
user@host# set thread-pool-size thread-pool-size
```

14. (Optional) Specify the timeout for stopping working threads after they become idle.

```
[edit shared sae configuration driver aaa]
user@host# set thread-idle-timeout thread-idle-timeout
```

15. (Optional) Configure the session store parameters for the AAA device driver.

    From configuration mode, access the configuration statement that configures the session store for the AAA device driver.

```
user@host# edit shared sae configuration driver aaa session-store
```

    For more information about configuring session store parameters, see Configuring the Session Store Feature (SRC CLI).

Configuring AAA Policies (SRC CLI)

Tasks to configure AAA policies are:

- Configuring AAA Policy Lists on page 143
- Configuring AAA Policy Rules on page 143
- Configuring Template Activation Actions on page 144

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 templates for RADIUS-enabled devices. You can configure template activation actions for AAA policy rules.

The template name and parameters are listed in the custom router template on the IMS AAA Server.

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**

CHAPTER 15

Monitoring the Diameter Server (SRC CLI)

SRC CLI Commands to Monitor the Diameter Server

You can view statistics and status for the Diameter server. Table 8 on page 147 lists the commands you use to monitor the Diameter server

Table 8: Commands to Monitor the Diameter Server

Command	Output Displayed
show diameter statistics	Information about the server process and the current state of the Diameter server.
show diameter statistics message-handler	Information about the Diameter server message handler.
show diameter statistics message-handler message-flow	Information about the Diameter server message flows.
show diameter statistics process	Information about the Diameter server process.
show diameter statistics requests	Information about the Diameter server requests.
show diameter status	Status of the Diameter server.
show diameter status clients	Status of the Diameter clients.
show diameter status peers	Status of the Diameter peers.

Viewing Statistics for the Diameter Server (SRC CLI)

Purpose View information about the server process and the state of the Diameter server.

Action To display information about the server process and the state of the Diameter server:

user@host> **show diameter statistics**

Viewing Message Handler Information for the Diameter Server (SRC CLI)

Purpose View information about the message handler and message flows for the Diameter server.

Action To display information about the message handler for the Diameter server:

user@host> **show diameter statistics message-handler**

To display information about message flows for the Diameter server:

user@host> **show diameter statistics message-handler message-flow**

To display information about a specific message flow:

user@host> **show diameter statistics message-handler message-flow id** *id*

Viewing Server Process Information for the Diameter Server (SRC CLI)

Purpose View information about the server process.

Action Purpose View information about the server process. Action To display about the server process:

user@host> **show diameter statistics process**

Viewing Information About Diameter Server Requests (SRC CLI)

Purpose View information about Diameter server requests.

Action To display information about Diameter server requests:

user@host> **show diameter statistics requests**

Viewing Diameter Server State (SRC CLI)

Purpose

Action To display information about the state of the Diameter server:

user@host> **show diameter status**

To display information about the Diameter clients:

user@host> **show diameter status clients**

To display information about a specific client:

user@host> **show diameter status clients client-name** *client-name*

To display information about the Diameter peers:

user@host> **show diameter status peers**

To display information about a specific peer:

user@host> **show diameter status peers peer-name** *peer-name*

CHAPTER 16

Managing Services with Diameter on MX Series Routers

Overview of SRC Peer Support on MX Series Routers

When the Juniper Networks routing platform supports the use of the Diameter protocol to provide extended AAA functionality, the SRC software can dynamically manage services on these devices. The SRC software uses the Diameter protocol for communications between the local SRC peer on a Juniper Networks routing platform, such as the Juniper Networks MX Series Ethernet Services Router, and the SAE. The local SRC peer is known as JSRC and is part of the AAA application.

JSRC has the following responsibilities:

- Request address authorization from the SAE.
- Request service activations from the SAE.
- Activate and deactivate services as specified by the SAE.
- Log out subscribers as specified by the SAE.
- Update the SAE with status of new service activations and deactivations.
- Synchronize subscriber state and service information with the SAE.
- Notify the SAE when subscribers log out.

The SRC software enables the SAE to activate and deactivate subscriber services and log out subscribers. The SAE can control only those resources that have been provisioned

through the SAE. Therefore, the SAE receives information about only those subscribers for whom JSRC has requested provisioning from the SAE. Similarly, the SAE can control only the subscriber services that it has activated.

Related Topics

Managing Services on MX Series Routers Using the Diameter Application

You can use the SRC software to manage services on Juniper Networks routing platforms using the Diameter protocol. The SRC software communicates with the local SRC peer on the device using Diameter messages to dynamically manage services for a subscriber session.

The SRC software includes a Diameter server that forwards AAR, ACR, SRQ, and STR messages from JSRC to the device driver in the SAE and that forwards PPR and ASR messages from the device driver to JSRC. These Diameter messages perform these functions:

- AA-Request (AAR)—Attach subscriber to access network
- Accounting-Request (ACR)—Provide accounting information
- Abort-Session-Request (ASR)—Disconnect subscriber
- Push-Profile-Request (PPR)—Start, modify, or stop service session
- Session-Resource-Query (SRQ)—Initiate synchronization
- Session-Termination-Request (STR)—Detach subscriber from access network

You configure the Diameter peers and a device for each device managed by the SAE. The Diameter server searches all devices of type junos-ise for virtual routers that include the local host in their SAE connections. For these devices, the Diameter server establishes a connection with the peers referenced in the device configuration.

Tasks to set up the management of services on devices using Diameter protocol:

- Configuring JSRC on the MX Series Router on page 152
- Configuring the Diameter Application (SRC CLI) on page 131
- Adding Network Devices (SRC CLI) on page 158
- Configuring Diameter Peers (SRC CLI) on page 137
- Configuring the SAE to Manage Network Devices (SRC CLI) on page 161
- Configuring JSRC Policies (SRC CLI) on page 162

Configuring JSRC on the MX Series Router

Tasks to set up JSRC on the Juniper Networks routing platform are:

- Configure the Diameter instance.
- Configure the local SRC peer, JSRC, for subscriber access.

- Configure subscribers over static interfaces.

For more information about JSRC and subscriber access, see the *Junos Subscriber Access Configuration Guide*.

Configuring the Diameter Application (SRC CLI)

Tasks to configure the Diameter application are:

Configuring the Diameter Application Properties

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;
   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 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) 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.

11. (Optional) Specify whether the peer connection is in debug mode.

    ```
    [edit system diameter]
    user@host# set debug-mode
    ```

12. (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)
    ```

13. (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.

14. (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.

15. (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.

16. (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).

Adding Network Devices (SRC CLI)

To set up the MX Series router so that it can be managed by the SAE:

1. From configuration mode, access the statements that configure network devices. You must specify the name of a device with lowercase characters. The name must match the Origin-Host AVP for the Diameter instance on the device. This sample procedure uses mx1 as the name of the router.

   user@host# edit shared network device mx1

2. Set the type of device to junos-ise.

   [edit shared network device mx1]
   user@host# set device-type junos-ise

3. Specify the configured peers associated with the device. See “Configuring Diameter Peers (SRC CLI)” on page 137.

   [edit shared network device mx1]
   user@host# set peers [*peers*...]

   Note that MX Series routers support only a single peer connection.

4. From configuration mode, access the statements for virtual routers. The name must match the JSRC partition configured on the MX Series router, which is configured within the logical system:routing instance context. This sample procedure uses the name * for the virtual router.

   [edit shared network device mx1]
   user@host# edit virtual-router *

   where * matches any JSRC partition. You can also specify that the JSRC partition be configured in a logical system or in a logical system and routing instance. By default, logical system **default** and routing instance **master** are used.

5. Specify the SAEs that can manage this router.

   [edit shared network device mx1 virtual-router default]
   user@host# set sae-connection [*sae-connection*...]

6. (Optional) Specify the VPN identifier used by this virtual router. You can specify VRF instead of a string to use the VRF instance reported by the device as the VPN identifier. In this case, the VPN identifier is the name of the routing instance.

   [edit shared network device mx1 virtual-router default]
   user@host# set vpn-id (*vpn-id* | VRF)

7. (Optional) Verify your configuration.

   ```
   [edit shared network device mx1]
   user@host# show
   ```

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 Topics

- Configuring the Diameter Application (SRC CLI) on page 131
- Managing Dynamic Policy Changes on MX Series Routers Using the Diameter Application on page 174
- Example: Configuring the SRC Software to Support PTSP on the MX Series Router on page 212

- Example: Configuring the SRC Software to Support Both PTSP and JSRC on the MX Series Router on page 214

Configuring the SAE to Manage Network Devices (SRC CLI)

Use the following configuration statements to configure the device driver for MX Series routers:

```
shared sae configuration driver junos-ise {
    sae-community-manager sae-community-manager;
    cached-driver-expiration cached-driver-expiration;
    keep-alive-timeout keep-alive-timeout;
    registry-retry-interval registry-retry-interval;
    reply-timeout reply-timeout;
    sequential-message-timeout sequential-message-timeout;
    thread-pool-size thread-pool-size;
    thread-idle-timeout thread-idle-timeout;
}
```

To configure the device driver:

1. From configuration mode, access the statements for the device driver.

   ```
   user@host# edit shared sae configuration driver junos-ise
   ```

2. Specify the name of the community manager.

   ```
   [edit shared sae configuration driver junos-ise]
   user@host# set sae-community-manager sae-community-manager
   ```

3. (Optional) Specify the minimum amount of time to keep the state of a device driver after its Diameter connection is closed.

   ```
   [edit shared sae configuration driver junos-ise]
   user@host# set cached-driver-expiration cached-driver-expiration
   ```

4. (Optional) Specify the keepalive timeout before the registry to a Diameter server expires.

   ```
   [edit shared sae configuration driver junos-ise]
   user@host# set keep-alive-timeout keep-alive-timeout
   ```

5. (Optional) Specify the interval between retrying a failed registry to a Diameter server.

   ```
   [edit shared sae configuration driver junos-ise]
   user@host# set registry-retry-interval registry-retry-interval
   ```

6. (Optional) Specify the timeout before a request sent to a Diameter server expires.

   ```
   [edit shared sae configuration driver junos-ise]
   user@host# set reply-timeout reply-timeout
   ```

7. (Optional) Specify the timeout before an expected message expires.

   ```
   [edit shared sae configuration driver junos-ise]
   user@host# set sequential-message-timeout sequential-message-timeout
   ```

8. (Optional) Specify the number of working threads that process requests.

```
[edit shared sae configuration driver junos-ise]
user@host# set thread-pool-size thread-pool-size
```

9. (Optional) Specify the timeout for stopping working threads after they become idle.

```
[edit shared sae configuration driver junos-ise]
user@host# set thread-idle-timeout thread-idle-timeout
```

10. (Optional) Configure the session store parameters for the device driver.

From configuration mode, access the statement that configures the session store for the device driver.

```
user@host# edit shared sae configuration driver junos-ise session-store
```

For more information about configuring session store parameters, see Configuring the Session Store Feature (SRC CLI).

Configuring JSRC Policies (SRC CLI)

Tasks to configure JSRC policies are:

- Configuring JSRC Policy Lists on page 162
- Configuring JSRC Policy Rules on page 162
- Configuring Dynamic Profile Actions on page 163

Configuring JSRC Policy Lists

To configure policy lists:

1. From configuration mode, create a policy list. For example, to create a policy list called l1 within a policy group called ise:

```
user@host# edit policies group ise list l1
```

2. Specify the type of policy list.

```
[edit policies group ise list l1]
user@host# set role junos-ise
```

3. Specify where the policy is applied on the device.

```
[edit policies group ise list l1]
user@host# set applicability both
```

Configuring JSRC Policy Rules

To configure 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 ise list l1 rule r1
```

2. Specify the type of policy rule.

```
[edit policies group ise list l1 rule r1]
user@host# set type junos-ise
```

Configuring Dynamic Profile Actions

Use this action to install existing dynamic profiles. You can configure dynamic profile actions for devices such as the MX Series routers.

The profile name must match a dynamic profile configured on the device and the variable name must match a variable configured for the dynamic profile.

Use the following configuration statements to configure a dynamic profile action:

```
policies group name list name rule name dynamic-profile name {
  profile-name profile-name;
  description description;
}
policies group name list name rule name dynamic-profile name variables name {
  value value;
  type type;
}
```

To configure a dynamic profile action:

1. From configuration mode, enter the dynamic profile action configuration. In this sample procedure, dp is the name of the dynamic profile action.

   ```
   user@host# edit policies group ise list l1 rule r1 dynamic-profile dp
   ```

2. Enter the profile name to activate.

   ```
   [edit policies group ise list l1 rule r1 dynamic-profile dp]
   user@host# set profile-name profile-name
   ```

3. (Optional) Enter a description for the dynamic profile action.

   ```
   [edit policies group ise list l1 rule r1 dynamic-profile dp]
   user@host# set description description
   ```

4. From configuration mode, enter the parameters used by the profile.

   ```
   user@host# edit policies group ise list l1 rule r1 dynamic-profile dp variables name
   ```

   For example:

   ```
   user@host# edit policies group ise list l1 rule r1 dynamic-profile dp variables
     upstreamBandwidth
   ```

5. (Optional) Configure the value for the variable.

   ```
   [edit policies group ise list l1 rule r1 dynamic-profile dp variables name]
   user@host# set value value
   ```

   For example:

   ```
   [edit policies group ise list l1 rule r1 dynamic-profile dp variables upstreamBandwidth]
   user@host# set value rateParameter
   ```

6. (Optional) Configure the variable type. Variable types are mapped to parameter types.

   [edit policies group ise list l1 rule r1 dynamic-profile dp variables ***name***]
   user@host# **set type *type***

   For example:

   [edit policies group ise list l1 rule r1 dynamic-profile dp variables upstreamBandwidth]
   user@host# **set type rate**

CHAPTER 17

Managing an MX Series Router as a Service Node

Overview of Service Nodes in an SRC Environment

The Juniper Networks MX Series Ethernet Services Router supports the packet-triggered subscribers and policy control (PTSP) feature that allows the dynamic application of policies on a per-subscriber basis to individual source IP addresses flowing through a given interface. A subscriber context is created for each distinct source IP address seen in a given underlying interface. This feature can be used to support subscribers who are controlled by a subscriber termination device, such as a B-RAS or GGSN device, that is connected to an MX Series router. MX Series routers that support PTSP are called service nodes.

Service nodes act as intelligent policy enforcement points for IP edge devices with these features:

- Single access–agnostic policy enforcement point that allows the easy introduction of new services independent of access technologies. Subscribers and policies are tracked by a subscriber's IP address and do not require subscriber interfaces.
- Single point for management, reporting, and troubleshooting, which includes support for dynamic policy attachment and updates.

When the MX Series router acts as a service node in the SRC environment, the SRC software supports this role by providing subscriber awareness to the service node.

To support service nodes, the SRC software:

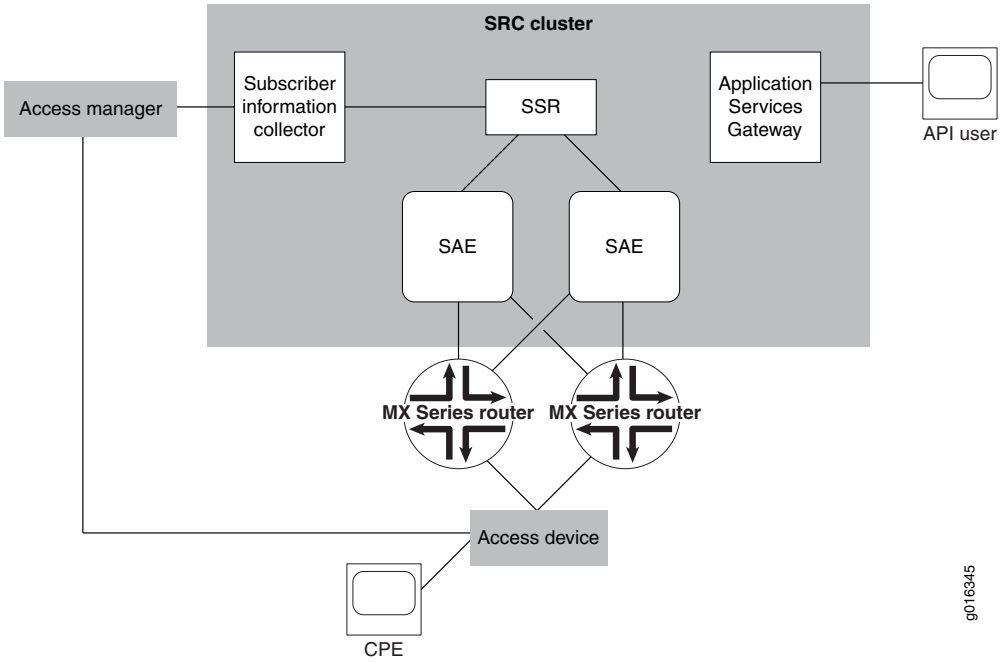
- Collects and dispatches RADIUS accounting events.
- Creates an IP edge attachment session and stores it.

- Manages profile and policies for IP address sessions (PTSP sessions) by associating the session with the correct attachment session and sending profile and policy information to the MX Series router.
- Sends start, interim, and stop accounting records containing usage information from the service node and attachment information from the IP edge device.

SRC Software in the Service Node Environment

Figure 16 on page 166 illustrates a simple deployment scenario.

Figure 16: SRC Software in the Service Node Environment



This simple deployment scenario includes the following components:

- Customer premises equipment (CPE)—Equipment with which the network user connects to an IP network. This device can be any device that allows a subscriber to connect to the network—including a wireless phone, a DSL router, or a cable modem.
- Access device—Device that terminates the IP session for the network user. This device must authenticate the network user and notify the access manager when an attachment session is created and stopped. Optionally, the device can notify the access manager when the attachment session is modified. This device can be a gateway GSN (GGSN), a Broadband Remote Access Server (B-RAS), or a cable modem termination system (CMTS).
- Access manager—Device that manages access devices. This device must be able to forward session start/stop notifications to the SRC cluster. This device can be a RADIUS accounting server.
- SRC cluster—Collection of SRC components that manage attachment sessions and PTSP device sessions, including:

  - Subscriber information collector (SIC)—SRC component that receives session start, modification, and stop notifications from the access manager. The start, modification, and stop notifications are RADIUS accounting start, interim update, and stop events.
  - Session State Registrar (SSR)—SRC component that stores attachment sessions and notifies other components about updates.
  - SAE—SRC component that manages service sessions and policies for IP subscriber sessions.
  - Application Services Gateway (ASG), API client—SRC cluster uses the gateway to allow external API clients to manipulate sessions maintained in the cluster.
- MX Series router—MX Series Ethernet Services Router that supports the PTSP feature (service node), which detects IP flows and manages policies for those sessions.

In this simple deployment scenario, the following actions might occur:

1. The CPE connects to an access device, which terminates an IP address and provides lifecycle notifications but provides limited policy management capabilities. The SRC cluster learns about CPE attachment sessions from the access manager.
2. Traffic from the CPE to the network is routed through the MX Series routers that support PTSP (service nodes).
3. The service node detects IP flows from the CPE and notifies the SRC cluster about the IP flow. Traffic from a single CPE can be routed through multiple MX Series routers; each router detects and manages individual flows that must be coordinated on the SRC cluster level.
4. The SRC cluster associates the IP flow information with identity information that it has learned for the CPE attachment session and uses this information to select the appropriate policies for handling the subscriber traffic.

Related Topics

- Subscriber Information Collector Overview
- Overview of the Session State Registrar
- Overview of Managing Subscriber-Level Policies on MX Series Routers on page 173

Using the SRC Software to Support PTSP

When you use the SRC software to support PTSP on MX Series routers, the SRC software can become aware of the subscribers before or after a PTSP session has been created. This topic describes the interaction among the components in the basic scenario and the sequence of events for different situations.

Accessing the Network Before the SRC Cluster Is Notified About a PTSP Session

The CPE connects to the network and the SRC cluster is notified about the connection before a PTSP session is created. In this case, the attachment session exists before the PTSP session. The sequence of events is:

1. The CPE connects to the network through the access device.
2. The access device notifies the access manager about the session start.
3. The access manager forwards the session start notification to the SIC, which translates the attributes into SRC-specific attributes.
4. The SIC creates an attachment session in the SSR.
5. (Optional) The subscriber activates a service through the ASG. (In the sequence of events, this step is the earliest one for using the ASG.)
6. (Optional) The ASG creates a service session in the SSR that is associated with the attachment session. If the attachment session does not exist, the service activation fails.
7. The CPE accesses the network. The service node detects a new IP flow and creates a PTSP session.
8. The service node notifies the SAE that is currently managing the MX Series router. The SAE extracts the IP address, and optionally the VPN ID, from the PTSP session information.
9. The SAE starts managing the PTSP session and calls the SSR reader authentication plug-in to obtain attachment session information from the SSR. The data from the SSR is used in the classification context.
10. The SAE runs the subscriber classification script, loads a subscriber profile, and creates a subscriber session. The subscriber session activates any subscribed activate-on-login service.
11. When the subscriber session is completely activated, the SAE installs any active policies on the service node.

Accessing the Network After the SRC Cluster Is Notified About a PTSP Session

The CPE connects to the network and the SRC cluster is notified about the connection after a PTSP session is processed. In this case, the attachment session does not exist before the PTSP session. The sequence of events is:

1. The CPE connects to the network through the access device.
2. The CPE accesses the network. The service node detects the new IP flow and creates a PTSP session.
3. The service node notifies the SAE that is currently managing the MX Series router. The SAE extracts the VPN ID from the PTSP session information.

4. The SAE starts managing the PTSP session and calls the SSR reader authentication plug-in to obtain attachment session information from the SSR. No information is returned to the SAE because the attachment session does not exist yet.
5. The SAE creates an unauthenticated (anonymous) subscriber session.
6. When the subscriber session is completely activated, the SAE installs any active policies on the service node.
7. The access device notifies the access manager about the session start.
8. The access manager forwards the session start notification to the SIC.
9. The SIC creates the attachment session in the SSR.
10. The SSR notifies the SAE that the attachment session has been modified. The SAE finds any affected subscriber sessions. The SSR notifies all active SAEs in the same cluster.
11. For any affected subscriber session, the SAE updates the classification context and initiates a login. This login runs the subscriber classification script and compares the result.
    - If the subscriber profile has changed, the existing session is terminated and a new session is created.
    - If the subscriber profile has not changed, the provisioned policies for active services are verified to determine whether they are affected by the updated attachment information.
12. Any changes are applied to the service node. If no policies have changed but the subscriber identity is different, the SAE changes the subscriber identity on the service node.
13. The subscriber activates a service through the ASG.
14. The ASG creates a service session in the SSR.
15. The SSR notifies all SAEs that a new service session exists.
16. The SAE that manages PTSP sessions for the attachment session activates a service session for the appropriate subscriber session.

Changing the Network Connection

The CPE connects to the network by different means. The attachment session is modified without changing the IP layer. The sequence of events is:

1. The CPE connects to the network through the access device in a different manner. For example, a wireless device roams to a different access point.
2. The access device notifies the access manager about the modified session parameters.
3. The access manager forwards the notification to the SIC.
4. The SIC updates the attachment session in the SSR.

5. The SSR notifies the SAE that the attachment session has been modified. The SAE finds any affected subscriber sessions. The SSR notifies all active SAEs in the same cluster.
6. For any affected subscriber session, the SAE updates the classification context and initiates a login. This login runs the subscriber classification script and compares the result.
   - If the subscriber profile has changed, the existing session is terminated and a new session is created.
   - If the subscriber profile has not changed, the provisioned policies for active services are verified to determine whether they are affected by the updated attachment information.
7. Any changes are applied to the service node. If no policies have changed but the subscriber identity is different, the SAE installs a policy on the service node with the changed subscriber identity.

If the IP layer is modified, the existing attachment session is terminated and a new attachment session is created.

Disconnecting from the Network

The network connection is terminated. The sequence of events for attachment session termination is:

1. The CPE disconnects from the network.
2. The access device notifies the access manager.
3. The access manager forwards the notification to the SIC.
4. The SIC terminates the attachment session in the SSR.
5. The SSR notifies the SAE.
6. The SAE terminates the subscriber session.
7. The SAE terminates the PTSP session on the service node.

Terminating the PTSP Session

The service node detects the end of the PTSP session because of an idle timeout. The sequence of events for PTSP session termination is:

1. The service node detects an idle timeout, terminates the PTSP session, and notifies the SAE.
2. The SAE terminates any subscriber session associated with the PTSP session, which terminates any service session and generates final accounting information.

PTSP session termination does not affect the attachment session.

If the attachment session remains active and the CPE accesses the network again, the sequence of events is the same as connecting to the network without a PTSP session.

If the attachment session terminates, the SAE receives a notification and terminates any remaining PTSP sessions associated with the attachment session. If there are no associated PTSP sessions, the SAE ignores the event.

Configuring SRC Software to Support Service Nodes

To configure the SRC components to support service nodes:

1. Configure the SRC software to communicate with the PTSP peer on the MX Series router, manage the MX Series router, and manage subscriber policies.

   See “Configuring PTSP to Manage Subscriber-Level Policies” on page 175.

2. Configure the SIC to listen for RADIUS accounting events from IP edge devices and filter events based on attachment session attributes.

   See Subscriber Information Collector Overview.

3. Configure the SSR to store information about IP edge attachment sessions.

   See Configuring the Initial SSR Cluster (SRC CLI).

4. (Optional) Configure Dynamic Service Activator methods for all subscriber sessions.

   See Enabling Dynamic Service Activator on a Web Application Server (SRC CLI).

CHAPTER 18

Managing Subscriber-Level Policies on MX Series Routers

Overview of Managing Subscriber-Level Policies on MX Series Routers

The Juniper Networks MX Series Ethernet Services Router supports a feature known as packet-triggered subscribers and policy control (PTSP). This feature allows dynamic policy and profile changes to be applied on a per-subscriber basis. The PTSP in the MX

Series router signals the SRC policy manager when a new IP address flow is detected or when an existing flow is idle, and allows the policy manager to dynamically apply new policies associated with the IP address flow.

The SRC software uses the Diameter protocol for communications between the local SRC peer on a Juniper Networks routing platform, such as the MX Series router, and the SAE. The local SRC peer in the MX Series router is known as the *PTSP*.

The PTSP device driver has the following responsibilities:

- Manage the logical connection to the MX Series router Diameter peer (PTSP).
- Receive IP address notifications from the MX Series router and dynamically activate, modify, or deactivate policies for existing subscriber sessions.
- Send subscriber identification information to the MX Series router.
- Push subscriber policy changes to MX Series router.
- Retrieve accounting information for active service sessions.
- Terminate a subscriber session.
- Synchronize the state of a single subscriber session or all sessions.

The PTSP device driver responds to requests from the MX Series router, which signals subscribers logging in and logging out. The driver publishes interface tracking events, performs interface classification to determine any default policies, and initiates SAE subscriber session login and logout processing.

Multiple instances of the device driver for the same device (MX Series router) can be configured in the network. The instances communicate with each other to provide redundancy. Only one driver for a given device is active at the same time.

The SAE activates, modifies and deactivates subscriber policies. The SAE can control only those resources that have been provisioned through the SAE. Therefore, the SAE receives information about only those subscribers (IP address flows) for whom PTSP has requested provisioning from the SAE. Similarly, the SAE can control only the subscriber policies that it has activated.

Related Topics

- Managing Dynamic Policy Changes on MX Series Routers Using the Diameter Application on page 174
- Policy Management Overview
- Policy Information Model

Managing Dynamic Policy Changes on MX Series Routers Using the Diameter Application

You can use the policy manager in the SRC software to dynamically manage subscriber level policy changes on MX Series routers. The PTSP device driver in the SRC software and the PTSP in the MX Series router are peers that communicate using a Diameter application. Information about subscriber policy changes is communicated between the PTSP and PTSP device driver using Diameter messages.

The SRC software includes a Diameter server process that forwards AAR, ACR, SRQ, and STR messages from PTSP to the PTSP device driver in the SAE, and that forwards PPR and ASR messages from the PTSP device driver to PTSP. These Diameter messages perform these functions:

- AA-Answer (AAA)—Sent in response to the AAR message for confirmation of the result of QoS provisioning.
- AA-Request (AAR)—Attaches the subscriber to the access network.
- Accounting-Request (ACR)—Provides accounting information.
- Abort-Session-Request (ASR)—Disconnects the subscriber.
- Push-Profile-Request (PPR)—Starts, modifies, or stops a service session.
- Session-Resource-Query (SRQ)—Initiates synchronization.
- Session-Termination-Request (STR)—Detaches the subscriber from the access network.

In the SRC software, you configure the Diameter peer (the MX Series router) and a device type (junos-ptsp) for each device managed by the SAE. The Diameter server process in the SRC software searches all devices of device type *junos-ptsp* for virtual routers that include the local host in their SAE connections. For these devices, the Diameter server process establishes a logical connection with the peers (in this case the MX Series router PTSP) referenced in the device configuration.

Related Topics

- Overview of Managing Subscriber-Level Policies on MX Series Routers on page 173
- Configuring the Diameter Application (SRC CLI) on page 131
- Configuring Diameter Peers (SRC CLI) on page 137

Configuring PTSP to Manage Subscriber-Level Policies

To configure PTSP to manage subscriber-level policies on the MX Series router:

- Configure PTSP on the MX Series router.

  See “Configuring PTSP on the MX Series Router” on page 176.
- Configure the Diameter application.

  See “Configuring the Diameter Application (SRC CLI)” on page 131
- Configure the Diameter peers.

  See “Configuring Diameter Peers (SRC CLI)” on page 137.
- Configure the SAE to manage PTSP on the MX Series router.

  See “Adding the MX Series Router as a PTSP Network Device (SRC CLI)” on page 183.
- Configure the SAE to obtain attachment session information from the SSR database.

  See “Configuring the SAE to Obtain Information About Subscribers (SRC CLI)” on page 185.

- Configure the PTSP device driver.

  See “Configuring the PTSP Device Driver (SRC CLI)” on page 186.

- Configure the session storage parameters for the PSTP device driver.

  See “Configuring the PTSP Device Driver Session Store (SRC CLI)” on page 187

- Configure the PTSP policies.

  See“Configuring PTSP Policies (SRC CLI)” on page 192.

Configuring PTSP on the MX Series Router

Tasks to set up PTSP on the Juniper Networks routing platform are:

- Configure the MultiServices DPC for PTSP.
- Configure the Diameter application to download dynamic PTSP policies.
- Configure any static PTSP policies.

Related Topics

- For more information about running PTSP on the MX Series router, see the *Junos Subscriber Access Configuration Guide*.
- Configuring PTSP to Manage Subscriber-Level Policies on page 175

Configuring the Diameter Application (SRC CLI)

Tasks to configure the Diameter application are:

- Configuring the Diameter Application Properties on page 176
- Configuring the Diameter Client Properties on page 179
- Configuring the Diameter Server Properties on page 180
- Configuring Logging Destinations on page 180

Configuring the Diameter Application Properties

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;
  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 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) 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.

11. (Optional) Specify whether the peer connection is in debug mode.

```
[edit system diameter]
user@host# set debug-mode
```

12. (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)
```

13. (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.

14. (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.

15. (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.

16. (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).

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 Topics

- Configuring the Diameter Application (SRC CLI) on page 131
- Managing Dynamic Policy Changes on MX Series Routers Using the Diameter Application on page 174
- Example: Configuring the SRC Software to Support PTSP on the MX Series Router on page 212
- Example: Configuring the SRC Software to Support Both PTSP and JSRC on the MX Series Router on page 214

Adding the MX Series Router as a PTSP Network Device (SRC CLI)

Use the following configuration statements to configure the network device:

```
shared network device name {
  device-type (junose | junos-ise | junos-ptsp | junos | pcmm | thirdparty);
  origin-host origin-host;
  peers [peers...];
}
shared network device name virtual-router name {
  sae-connection [sae-connection...];
  authentication-plug-in [authentication-plug-in...];
  vpn-id vpn-id;
}
```

To configure the MX Series router PTSP so that it can be managed by the SAE:

1. From configuration mode, access the statements that configure network devices. You must specify the name of a device with lowercase characters. This sample procedure uses mx-name as the name of the router.

   ```
   user@host# edit shared network device mx-name
   ```

2. Set the type of device to junos-ptsp.

   ```
   [edit shared network device mx-name]
   user@host# set device-type junos-ptsp
   ```

3. (Optional) Specify the origin host name of the MX Series router. This example procedure uses mx-origin-host as the origin host name. The active SAE registers events from the router based on the configured origin host attribute (mx-origin-host in the example). If the origin host is not configured, SAE uses the device name (mx-name in the example) instead.

   ```
   user@host# edit shared network origin-host mx-origin-host
   ```

4. Specify the configured peers associated with the device. See “Configuring Diameter Peers (SRC CLI)” on page 137.

   ```
   [edit shared network device mx-name]
   user@host# set peers [peers...]
   ```

   Note that MX Series routers support only a single peer connection.

5. From configuration mode, access the statements for virtual routers. The name must match the PTSP partition configured on the MX Series router, which is configured within the logical system:routing instance context. This sample procedure uses the name * for the virtual router.

   ```
   [edit shared network device mx-name]
   user@host# edit virtual-router *
   ```

   where * matches any PTSP partition. You can also specify that the PTSP partition be configured in a logical system or in a logical system and routing instance. By default, logical system **default** and routing instance **master** are used.

6. Specify the SAEs that can manage this router.

   ```
   [edit shared network device mx-name virtual-router default]
   user@host# set sae-connection [sae-connection...]
   ```

7. (Optional) Specify the plug-ins that authenticate subscribers who log in through this virtual router.

   ```
   [edit shared network device mx-name virtual-router default]
   user@host# set authentication-plug-in [authentication-plug-in...]
   ```

8. (Optional) Specify the VPN identifier used by this virtual router. You can specify VRF instead of a string to use the VRF instance reported by the device as the VPN identifier. In this case, the VPN identifier is the name of the routing instance.

   ```
   [edit shared network device mx-name virtual-router default]
   user@host# set vpn-id (vpn-id | VRF)
   ```

9. (Optional) Verify your configuration.

   ```
   [edit shared network device mx-name]
   user@host# show
   ```

Related Topics

- Configuring Diameter Peers (SRC CLI) on page 137
- Adding JUNOS Routing Platforms and Virtual Routers (SRC CLI)
- Adding Objects for Network Devices (SRC CLI)
- Configuring the SAE to Manage JUNOS Routing Platforms (SRC CLI)
- Configuring the SAE to Obtain Information About Subscribers (SRC CLI) on page 185
- Configuring the PTSP Device Driver (SRC CLI) on page 186

Configuring the SAE to Obtain Information About Subscribers (SRC CLI)

You can configure the SAE to obtain information about dynamic subscribers from the MX Series routers that support PTSP by making attachment session information available to the SAE. Tasks to configure the SAE for this purpose are:

Obtaining Subscriber Session Information from the SSR Database

You can obtain subscriber information from the Session State Registrar (SSR) database. The SSR reader plug-in obtains information about attachment sessions from the SSR to set the values for the plug-in attributes. The SSR reader authentication plug-in can be used by a specific virtual router or for all virtual routers.

Use the following configuration statements to set up an SSR reader plug-in:

```
shared sae configuration plug-ins name name ssr-reader {
    read-attributes [read-attributes...];
}
```

To configure the SSR reader plug-in:

1. From configuration mode, access the SSR reader plug-in configuration.

   user@host# edit shared sae configuration plug-ins name *name* ssr-reader

2. (Optional) Specify the plug-in attribute whose value is set from the SSR subscriber sessions table.

   [edit shared sae configuration plug-ins name *name* ssr-reader]
   user@host# set read-attributes *[read-attributes...]*

Configuring Event Publishers

You can configure the event publisher to use the SSR reader authentication plug-in.

To configure the global or default event publisher for PTSP:

1. From configuration mode, access the statement that configures the event publisher.

   user@host# edit shared sae configuration plug-ins event-publishers

2. (Optional) Specify the plug-ins that authenticate subscribers who are assigned to a virtual router that does not specify an authentication plug-in.

   [edit shared sae configuration plug-ins event-publishers]
   user@host# set default-vr-authentication *[default-vr-authentication...]*

3. (Optional) Specify the plug-ins that authenticate subscribers who log in through a specific type of device (for example, junos-ptsp).

   [edit shared sae configuration plug-ins event-publishers]
   user@host# set device-type-authentication junos-ptsp *[plug-ins...]*

To specify the plug-ins that authenticate subscribers who log in through a specific virtual router, set the **authentication-plug-in** option when you are adding the virtual router for the network device. See "Adding the MX Series Router as a PTSP Network Device (SRC CLI)" on page 183.

Configuring the PTSP Device Driver (SRC CLI)

In most cases, all attributes have reasonable defaults and should not require configuration. The configuration should be changed only by advanced users wishing to tune the performance.

Use the following configuration statements to configure the PTSP device driver for MX Series routers:

```
shared sae configuration driver junos-ptsp {
    sae-community-manager  sae-community-manager;
    cached-driver-expiration cached-driver-expiration;
    keep-alive-timeout keep-alive-timeout;
    registry-retry-interval registry-retry-interval;
    reply-timeout reply-timeout;
    sequential-message-timeout  sequential-message-timeout;
    thread-pool-size  thread-pool-size;
    thread-idle-timeout thread-idle-timeout;
}
```

To configure the device driver:

1. From configuration mode, access the statements for the device driver.

   ```
   user@host# edit shared sae configuration driver junos-ptsp
   ```

2. Specify the name of the community manager.

   ```
   [edit shared sae configuration driver junos-ptsp]
   user@host# set sae-community-manager sae-community-manager
   ```

3. (Optional) Specify the minimum amount of time to keep the state of a device driver after its Diameter connection is closed.

   ```
   [edit shared sae configuration driver junos-ptsp]
   user@host# set cached-driver-expiration cached-driver-expiration
   ```

4. (Optional) Specify the keepalive timeout before the registry to a Diameter server expires.

   ```
   [edit shared sae configuration driver junos-ptsp]
   user@host# set keep-alive-timeout keep-alive-timeout
   ```

5. (Optional) Specify the interval between retrying a failed registry to a Diameter server.

   ```
   [edit shared sae configuration driver junos-ptsp]
   user@host# set registry-retry-interval registry-retry-interval
   ```

6. (Optional) Specify the length of time before a request sent to a Diameter server expires.

   ```
   [edit shared sae configuration driver junos-ptsp]
   user@host# set reply-timeout reply-timeout
   ```

7. (Optional) Specify the length of time before an expected message expires.

   ```
   [edit shared sae configuration driver junos-ptsp]
   user@host# set sequential-message-timeout sequential-message-timeout
   ```

8. (Optional) Specify the number of working threads that process requests.

   ```
   [edit shared sae configuration driver junos-ptsp]
   user@host# set thread-pool-size thread-pool-size
   ```

9. (Optional) Specify the length of time for stopping working threads after they become idle.

   ```
   [edit shared sae configuration driver junos-ptsp]
   user@host# set thread-idle-timeout thread-idle-timeout
   ```

10. (Optional) Configure the session store parameters for the device driver.

    From configuration mode, access the statement that configures the session store for the device driver.

    ```
    user@host# edit shared sae configuration driver junos-ptsp session-store
    ```

    For more information about configuring session store parameters, see Configuring the Session Store Feature (SRC CLI).

Related Topics

- Configuring the SAE to Manage JUNOS Routing Platforms (SRC CLI)
- Configuring the PTSP Device Driver Session Store (SRC CLI) on page 187

Configuring the PTSP Device Driver Session Store (SRC CLI)

In most cases, all attributes have reasonable defaults and should not require configuration. The configuration should be changed only by advanced users wishing to tune the performance.

Use the following configuration statements to configure the PTSP device driver session storage configuration:

```
shared sae configuration driver junos-ptsp session-store {
    maximum-queue-age maximum-queue-age;
    maximum-queued-operations maximum-queued-operations;
    maximum-queue-size maximum-queue-size;
    maximum-file-size maximum-file-size;
    minimum-disk-space-usage minimum-disk-space-usage;
    rotation-batch-size rotation-batch-size;
    maximum-session-size maximum-session-size;
    disk-load-buffer-size disk-load-buffer-size;
    network-buffer-size network-buffer-size;
    retry-interval retry-interval;
    communications-timeout communications-timeout;
    load-timeout load-timeout;
    idle-timeout idle-timeout;
    maximum-backlog-ratio maximum-backlog-ratio;
    minimum-backlog minimum-backlog;
}
```

To configure the PTSP device driver session storage:

1. From configuration mode, access the statements for the driver session storage.

   user@host# edit shared sae configuration driver junos-ptsp session-store

2. (Optional) Specify the maximum age that a queue of buffered store operations (such as adding a session to the store or removing a session from the store) can reach before the queue is written to a session store file.

   [edit shared sae configuration driver junos-ptsp session-store]
   user@host# set maximum-queue-age ***maximum-queue-age***

   Enter a value for the number of milliseconds in the range 0–2147483647. A value of –1 indicates that there is no limit. A value of 0 causes the session store to write each store operation to a session store file immediately.

3. (Optional) Specify the number of buffered store operations that are queued before the queue is written to a session store file.

   [edit shared sae configuration driver junos-ptsp session-store]
   user@host# set maximum-queued-operations ***maximum-queued-operations***

   Enter an integer in the range 0–2147483647. A value of –1 indicates that there is no limit. A value of 0 causes the session store to write each store operation to a session store file immediately.

4. (Optional) Specify the maximum size that a queue of buffered store operations can reach before the queue is written to a session store file.

   [edit shared sae configuration driver junos-ptsp session-store]
   user@host# set maximum-queue-size ***maximum-queue-size***

   Enter the number of bytes in the range 0–2147483647.

5. (Optional) Specify the maximum size of session store files. When a file reaches this size, a new file is created.

   [edit shared sae configuration driver junos-ptsp session-store]
   user@host# set maximum-file-size ***maximum-file-size***

   Enter the number of bytes in the range 0–2147483647.

6. (Optional) Specify the percentage of space in all session store files that is used by live sessions. When the space in the session store files that is used by live sessions decreases to this percentage, the oldest session store file is compacted and appended to the newest session store file, and then the oldest session store file is deleted.

   [edit shared sae configuration driver junos-ptsp session-store]
   user@host# set minimum-disk-space-usage ***minimum-disk-space-usage***

   Enter a percentage of disk space in the range 1–100. We recommend a range of 30-50.

7. (Optional) Specify when the oldest session store file is rotated. The value specifies the number of sessions that are rotated from the oldest file to the newest file at the same time. While a set of sessions is rotated, no other session store activity can take place

```
[edit shared sae configuration driver junos-ptsp session-store]
user@host# set rotation-batch-size rotation-batch-size
```

Enter an integer in the range 0–2147483647.

8. (Optional) Specify the maximum size of a single subscriber or service session. Use this parameter to reserve memory for an internal buffer.

```
[edit shared sae configuration driver junos-ptsp session-store]
user@host# set maximum-session-size maximum-session-size
```

Enter the number of bytes in the range 0–2147483647.

9. (Optional) Specify the size of the buffer that is used to load all of a session store's files from disk at startup.

```
[edit shared sae configuration driver junos-ptsp session-store]
user@host# set disk-load-buffer-size disk-load-buffer-size
```

Enter the number of bytes in the range 0–2147483647.

10. (Optional) Specify the size of the buffer that holds messages or message segments that are waiting to be sent to passive session stores.

```
[edit shared sae configuration driver junos-ptsp session-store]
user@host# set network-buffer-size network-buffer-size
```

The number of bytes entered must be larger than or equal to 21 plus *maximum-session-size* and less than 2147483647.

11. (Optional) Specify the time interval to be allowed between attempts by the active session store to connect to missing passive session stores.

```
[edit shared sae configuration driver junos-ptsp session-store]
user@host# set retry-interval retry-interval
```

Enter the number of milliseconds in the range 0–2147483647.

12. (Optional) Specify the amount of time in milliseconds that a session store should wait before closing when it is blocked from reading or writing a message. This timeout does not apply when a session store is waiting for a remote session store to load its state from disk.

```
[edit shared sae configuration driver junos-ptsp session-store]
user@host# set communications-timeout communications-timeout
```

Enter the number of milliseconds. (A nonpositive number means wait forever. This is not recommended.)

13. (Optional) Specify the amount of time in milliseconds that an active session store should wait for a passive session store or a passive session store waits for an active session store to load its data from disk before it closes the connection to the session store.

```
[edit shared sae configuration driver junos-ptsp session-store]
user@host# set load-timeout load-timeout
```

Enter the number of milliseconds. (A nonpositive number means wait forever. This is not recommended.)

14. (Optional) Specify the amount of time that a passive session store waits for activity from the active session store before it closes the connection to the active session store. This timeout applies after the session store startup and initial update processes are complete.

    ```
    [edit shared sae configuration driver junos-ptsp session-store]
    user@host# set idle-timeout idle-timeout
    ```

    Enter the number of milliseconds in the range 0–2147483647.

15. (Optional) Specify the maximum backlog ratio. Along with the minimum backlog size, this ratio specifies when the active session store closes the connection to a passive session store because of a backlog of messages waiting to be sent. After the startup and initial update processes are complete, if the backlog becomes too large, the connection to the passive session store is closed. After the retry interval ends, a new connection is opened.

    If the backlog of unsent operations (in bytes) divided by the total size (in bytes) of all live store operations is greater than this number, the connection is closed.

    ```
    [edit shared sae configuration driver junos-ptsp session-store]
    user@host# set maximum-backlog-ratio maximum-backlog-ratio
    ```

    Enter a floating point number.

16. (Optional) Specify the size of the minimum backlog. Along with the maximum backlog ratio, this number specifies when the active session store closes the connection to a passive session store because of a backlog of messages waiting to be sent to the passive session store. After the startup and initial update processes are complete, if the backlog becomes too large, the connection to the passive session store is closed. After the retry interval ends, a new connection is opened. If the maximum backlog ratio is met, the active session store does not close the connection unless the backlog of messages (in bytes) is greater than this number.

    ```
    [edit shared sae configuration driver junos-ptsp session-store]
    user@host# set minimum-backlog minimum-backlog
    ```

    Enter the number of bytes in the range 0–2147483647.

Related Topics

- Configuring PTSP Policies (SRC CLI) on page 192
- Overview of Managing Subscriber-Level Policies on MX Series Routers on page 173

Configuration Statements for PTSP Policies (SRC CLI)

Use the following configuration statements to configure PTSP policies:

```
policies group name {
  description description ;
}
policies group name list name {
  role [(junos | junose-ipv4 | junose-ipv6 | junose-l2tp | pcmm | aaa | junos-ise | junos-ptsp)];
  applicability [(input | output | both | secondary-input)];
  description description;
}
```

```
policies group name list name rule name {
  precedence precedence;
  accounting;
  application-accounting application-accounting;
  type [(ptsp-service-rule | ptsp-template)];
}
policies group name list name policer name {
  bandwidth bandwidth;
  max-burst-size max-burst-size;
}
policies group name list name rule name traffic-condition name {
  match-direction match-direction;
  description description;
}
policies group name list name rule name traffic-condition name destination-network
  network {
  ip-address ip-address;
  ip-mask ip-mask;
}
policies group name list name rule name traffic-condition name destination-network
  group-network {
  network-specifier network-specifier;
}
policies group name list name rule name traffic-condition name protocol-condition {
  protocol protocol;
}
policies group name list name rule name traffic-condition name protocol-port-condition
  {
  protocol protocol;
}
policies group name list name rule name traffic-condition name protocol-port-condition
  destination-port port {
  from-port from-port;
}
policies group name list name rule name traffic-condition name protocol-port-condition
  source-port port {
  from-port from-port;
}
policies group name list name rule name traffic-condition name
  parameter-protocol-condition {
  protocol protocol;
}
policies group name list name rule name traffic-condition name
  parameter-protocol-condition proto-attr destination-port port {
  from-port from-port;
}
policies group name list name rule name traffic-condition name
  parameter-protocol-condition proto-attr source-port port {
  from-port from-port;
}
policies group name list name rule name traffic-condition name tcp-condition {
  protocol protocol;
}
policies group name list name rule name traffic-condition name tcp-condition
  destination-port port {
  from-port from-port;
```

```
}
policies group name list name rule name traffic-condition name tcp-condition
  source-port port {
  from-port from-port;
}
policies group name list name rule name traffic-condition name traffic-match-condition
  {
  term-precedence term-precedence
  application [application...];
  application-group [application-group...];
}
policies group name list name rule name policer-ref {
  policer-name;
  description description;
}
policies group name list name rule name forwarding-instance {
  forwarding-instance;
  forwarding-unit forwarding-unit ;
  description description ;
}
policies group name list name rule name forwarding-class name {
  forwarding-class forwarding-class ;
  description description ;
}
policies group name list name rule name filter name {
  description description ;
}
```

Configuring PTSP Policies (SRC CLI)

The role of the policy list for the PTSP device driver must be set to *junos-ptsp*. The policy list must be configured to contain the rule of type *ptsp-service-rule*. A ptsp-service-rule object can contain one or more traffic-conditions and any number of actions (filter, policer-ref, forwarding-class, forwarding-instance). Each traffic-condition is translated to the PTSP template policy *__svc_rule__* using the same action variables defined in the rule.

Before you configure PTSP policies, review the information about configuring and managing policies:

- Policy Management Overview
- Policy Information Model
- Before You Configure SRC Policies
- Enabling the Policy Configuration on the SRC CLI

To configure PTSP policies:

1. Create a policy group.

   See “Configuring Policy Groups (SRC CLI)” on page 193.

2. Configure the policy list and set the **role** of the list to **junos-ptsp** and the **applicability** to **both**.

See “Configuring PTSP Policy Lists (SRC CLI)” on page 194.

3. Configure the PTSP policer instance.

   See “Configuring the PTSP Policer Instance (SRC CLI)” on page 194.

4. Configure the PTSP policy rule and set the rule **type** to **ptsp-service-rule**.

   See “Configuring PTSP Policy Rules (SRC CLI)” on page 196.

5. Configure the PTSP classify-traffix conditions.

   See “Configuring PTSP Classify-Traffic Conditions (SRC CLI)” on page 197.

6. Configure the PTSP actions.

   See “Configuring PTSP Actions” on page 209.

Related Topics

- Policy Components
- Configuration Statements for PTSP Policies (SRC CLI) on page 190

Configuring Policy Groups (SRC CLI)

Policy groups hold policy lists. You can create policy groups within policy folders. Use the following configuration statement to create a policy group:

```
policies group name {
  description description ;
}
```

To create a policy group:

1. From configuration mode, enter the **edit policies group** statement. For example, to create a folder called dhcp-default:

   ```
   user@host# edit policies group dhcp-default
   ```

2. (Optional) Enter a description for the policy group.

   ```
   [edit policies group dhcp-default]
   user@host# set description description
   ```

3. (Optional) Verify your policy group configuration.

   ```
   [edit policies group dhcp-default]
   user@host#show
   description "Default policy for JUNOSe routers";
   ```

Related Topics

- Before You Configure SRC Policies
- Configuring Policy Folders (SRC CLI)
- Enabling the Policy Configuration on the SRC CLI
- Configuring Policy Groups (C-Web Interface)

- Example: Creating Access Policies for Subscribers

Configuring PTSP Policy Lists (SRC CLI)

NOTE: For PTSP you must:

- Set the role of the policy list to junos-ptsp
- Set the policy list rule type to ptsp-service-rule
- Set the policy list applicability option to both
- Create a policer instance.

Use the following configuration statements to PTSP policy lists:

```
policies group name list name {
    role [(junos | junose-ipv4 | junose-ipv6 | junose-l2tp | pcmm | aaa | junos-ise | junos-ptsp)];
    applicability [(input | output | both | secondary-input)];
    description description;
}
```

To configure policy lists:

1. From configuration mode, create a policy list. For example, to create a policy list called list1 within a policy group called group1:

   ```
   user@host# edit policies group group1 list list1
   ```

2. Specify the role of the policy list. For PTSP the role must be set to **junos-ptsp**.

   ```
   [edit policies group group1 list list1]
   user@host# set role junos-ptsp
   ```

3. Specify where the policy is applied on the device. For PTSP the **applicability** option must be set to **both**.

   ```
   [edit policies group group1 list list1]
   user@host# set applicability both
   ```

Related Topics

- Policy Management Overview
- Configuring PTSP Policies (SRC CLI) on page 192
- Configuring PTSP Policy Rules (SRC CLI) on page 196
- Configuring the PTSP Policer Instance (SRC CLI) on page 194

Configuring the PTSP Policer Instance (SRC CLI)

Optionally, configure one or more policer instances that can be referenced by one or more PTSP policer-ref actions. The policer instance can be shared by different service

rules inside the same policy list. If the policer instance is shared, all packets matching any of the service rules are policed together.

NOTE: You need to configure a policer instance only if a policy rule references the policer.

NOTE: For PTSP you must:

- Set the role of the policy list to junos-ptsp
- Set the policy list rule type to ptsp-service-rule
- Set the policy list applicability option to both
- Create a policer instance.

Use the following configuration statements to configure the policer instance:

```
policies group name list name policer name {
  bandwidth bandwidth;
  max-burst-size max-burst-size;
}
```

To configure a policer instance:

1. (Optional) From configuration mode, create a policer instance. In this example the policer instance is called policer1.

   ```
   user@host# edit policies group name list name policer policer1
   ```

2. (Optional) Specify the bandwidth for the policer instance.

   ```
   [edit policies group name list name policer policer1]

   user@host# set bandwidth bandwidth
   ```

   Enter an integer between 8000–40000000000 bits per second.

3. (Optional) Specify the maximum burst size for the policer instance.

   ```
   [edit policies group name list name policer policer1]
   user@host# set max-burst-size max-burst-size
   ```

   Enter an integer between 1500–100000000000 octets.

Related Topics

- Configuring PTSP Policies (SRC CLI) on page 192
- Configuring PTSP Actions on page 209
- Configuring PTSP Policy Lists (SRC CLI) on page 194
- Configuring PTSP Policy Rules (SRC CLI) on page 196
- Configuring PTSP Classify-Traffic Conditions (SRC CLI) on page 197

Configuring PTSP Policy Rules (SRC CLI)

NOTE: For PTSP you must:

- Set the role of the policy list to junos-ptsp
- Set the policy list rule type to ptsp-service-rule
- Set the policy list applicability option to both
- Create a policer instance.

Use the following configuration statements to configure PTSP policy rules:

```
policies group name list name rule name {
    precedence precedence;
    accounting;
    application-accounting application-accounting;
    type [(ptsp-service-rule | ptsp-template)];
}
```

To configure 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 rule1 within a policy list called list1:

   ```
   user@host# edit policies group group1 list list1 rule rule1
   ```

2. (Optional) Specify the order in which the policy manager applies the policy rule. Rules are evaluated from lowest to highest precedence value. Precedence has meaning only if two rules have different classifiers and if those classifiers overlap. If this is the case and a packet is received that satisfies both classifiers, then only the action of the rule with the lower precedence value is performed.

   For PTSP policies, enter an integer in the range 1–254.

   ```
   [edit policies group group1 list list1 rule rule1]
   user@host# set precedence  precedence
   ```

3. (Optional) Specify whether accounting data is collected for the actions in the policy rule.

   If you specify that accounting data is collected, the SAE begins collecting accounting information when a service that uses the policy rule is activated. When the service is deactivated, the SAE sends the accounting records to the RADIUS accounting server or to a plug-in.

   When you specify multiple actions for accounting, the SAE adds the accounting data for individual actions together to obtain a summary accounting record for that interface direction.

   Accounting is not available for all actions.

```
[edit policies group group1 list list1 rule rule1
user@host# set accounting
```

4. (Optional) Specify application accounting.

   If PTSP application accounting is configured on the MX Series router, this attribute selects how application accounting is collected. Application accounting is maintained in a flat file on the router and is not collected by the SRC software. Application accounting and rule accounting are mutually exclusive.

   ```
   [edit policies group group1 list list1 rule rule1
   user@host# set application-accounting application-accounting
   ```

   Application accounting can be set to one of the following values:

   - **application**—Router maintains one counter per subscriber/application.
   - **group**—Router maintains one counter per subscriber/application group.
   - **any**—Router maintains one counter per subscriber.
   - A parameter of type applicationAccounting.

5. Specify the type of policy rule.

   ```
   [edit policies group group1 list list1 rule rule1
   user@host# set type type
   ```

   Where ***type*** is one of the following values:

   - **ptsp-service-rule**—Use the predefined policy template called __svc_rule__ on the MX Series router.
   - **ptsp-template**—This setting is for future use.

Related Topics

- Policy Management Overview
- Configuring PTSP Policies (SRC CLI) on page 192
- Configuring the PTSP Policer Instance (SRC CLI) on page 194
- Configuring PTSP Actions on page 209
- Configuring PTSP Classify-Traffic Conditions (SRC CLI) on page 197

Configuring PTSP Classify-Traffic Conditions (SRC CLI)

Before you configure PTSP classify-traffic conditions, review the following topics:

- Policy Management Overview
- Policy Components
- Policy Information Model

Topics that discuss configuring PTSP classify-traffic conditions include:

Creating PTSP Classify-Traffic Conditions (SRC CLI)

You create classify-traffic conditions within policy rules. Use the following configuration statements to create a classify-traffic condition:

```
policies group name list name rule name traffic-condition name {
  match-direction match-direction;
  description  description;
}
```

To add a classify-traffic condition:

1. From configuration mode, create a classify-traffic condition inside a policy rule that has already been created and configured. For example, to create a traffic-condition called condition1 within policy rule rule1:

   ```
   user@host# edit policies group group1 list list1 rule rule1 traffic-condition condition1
   ```

2. (Optional) Specify the direction of the packet flow on which you want to match packets.

   ```
   [edit policies group group1 list list1 rule rule1 traffic-condition condition1]
   user@host# set match-direction match-direction
   ```

   Set to one of the following values:

   - input
   - output
   - both
   - Parameter of type matchDirection

3. (Optional) Provide a description of the classify-traffic condition.

```
[edit policies group group1 list list1 rule rule1 traffic-condition condition1]
user@host# set description description
```

4. (Optional) Verify your PTSP classify-traffic condition configuration.

```
[edit policies group group1 list list1 rule rule1 traffic-condition
condition1]
user@host# show
match-direction output;
description "Destination classifier";
```

Configuring Destination Networks for PTSP Classify-Traffic Conditions (SRC CLI)

Use the following configuration statements to add destination networks to a PTSP classify-traffic condition:

```
policies group name list name rule name traffic-condition name destination-network
  network {
  ip-address ip-address;
  ip-mask ip-mask;
}
```

To add a destination network to a PTSP classify-traffic condition:

1. From configuration mode, enter the destination network within a classify-traffic condition. For example:

```
user@host# edit policies group group1 list list1 rule rule1 traffic-condition condition1
  destination-network network
```

2. (Optional) Specify the IP address of the destination network or host.

```
[edit policies group group1 list list1 rule rule1 traffic-condition condition1
  destination-network network]
user@host# set ip-address ip-address
```

Where ***ip-address*** is one of the following values:

- IP address
- Predefined global parameter:
  - gateway_ipAddress—IP address of the gateway as specified by the service object.
  - interface_ipAddress—IP address of the router interface.
  - service_ipAddress— IP address of the service as specified by the service object .
  - user_ipAddress—IP address of the subscriber.
  - virtual_ipAddress—Virtual portal address of the SAE that is used in redundant redirect server installations.
- Parameter of type address

3. (Optional) Configure the IP mask of the destination network or host.

   [edit policies group group1 list list1 rule rule1 traffic-condition condition1 destination-network network]
   user@host# set ip-mask *ip-mask*

   Where ***ip-mask*** is one of the following values:

   - IP address mask
   - Predefined global parameter:
     - interface_ ipMask—IP mask of the router interface.
     - service_ ipMask—IP mask of the service as specified by the service object.
     - user_ ipMask—IP mask of the subscriber.
   - Parameter of type address.

4. (Optional) Verify your destination network configuration.

```
[edit policies group group1 list list1 rule rule1 traffic-condition
condition1 destination-network network]
user@host# show
ip-address interface_ipAddress;
ip-mask interface_ipMask;
```

Configuring Destination Grouped Networks for PTSP Classify-Traffic Conditions (SRC CLI)

Use the following configuration statements to add destination networks in a grouped format to a classify-traffic condition:

```
policies group name list name rule name traffic-condition name destination-network
  group-network {
  network-specifier network-specifier;
}
```

To add a grouped destination network to a classify-traffic condition:

1. From configuration mode, enter the destination network within a classify-traffic condition. For example:

   user@host# edit policies group group1 list list1 rule rule1 traffic-condition condition1 destination-network group-network

2. (Optional) Configure the IP address of the destination network or host.

   [edit policies group group1 list list1 rule rule1 traffic-condition condition1 destination-network group-network]
   user@host# set network-specifier *network-specifier*

3. (Optional) Verify your destination network configuration.

```
[edit policies group group1 list list1 rule rule1 traffic-condition
condition1 destination-network group-network]
user@host# show
network-specifier any;
```

Configuring Protocol Conditions for PTSP Classify-Traffic Conditions (SRC CLI)

The procedure in this topic shows how to configure protocol conditions that do not include port conditions.

- If your condition includes port numbers, use the procedure in "Configuring Protocol Conditions with Ports for PTSP Classify-Traffic Conditions (SRC CLI)" on page 201.
- If your condition consists of a protocol that is assigned with a parameter value, use the procedure in "Configuring Protocol Conditions with Parameters for PTSP Classify-Traffic Conditions (SRC CLI)" on page 204.

Use the following configuration statements to add general protocol conditions to a PTSP classify-traffic condition:

```
policies group name list name rule name traffic-condition name protocol-condition {
  protocol protocol;
}
```

To add general protocol conditions to a classify-traffic condition:

1. From configuration mode, enter the general protocol condition configuration. For example:

   ```
   user@host# edit policies group group1 list list1 rule rule1 traffic-condition condition1
     protocol-condition
   ```

2. Configure the protocol matched by this classify-traffic condition.

   ```
   [edit policies group group1 list list1 rule rule1 traffic-condition condition1
     protocol-condition]
   user@host# set protocol protocol
   ```

   Enter the protocol matched by this classifier list, one of the following values:

   - Predefined global parameter—Use a ? at the command line to see a list of valid protocols.
   - Protocol number in the range 0–255.
   - String expression.
   - Parameter of type protocol.

3. (Optional) Verify your protocol condition configuration.

   ```
   [edit policies group group1 list list1 rule rule1 traffic-condition
   condition1 protocol-condition]
   user@host# show
   protocol 0;
   ```

Configuring Protocol Conditions with Ports for PTSP Classify-Traffic Conditions (SRC CLI)

Use the following configuration statements to add general protocol conditions with ports to a PTSP classify-traffic condition:

```
policies group name list name rule name traffic-condition name protocol-port-condition
  {
  protocol  protocol;
}
policies group name list name rule name traffic-condition name protocol-port-condition
  destination-port port {
  from-port from-port;
}
policies group name list name rule name traffic-condition name protocol-port-condition
  source-port port {
  from-port from-port;
}
```

To add general protocol conditions with ports to a PTSP classify-traffic condition:

1. From configuration mode, enter the protocol port condition configuration. For example:

   ```
   user@host# edit policies group group1 list list1 rule rule1 traffic-condition condition1
     protocol-port-condition
   ```

2. Configure the protocol matched by this classify-traffic condition.

   ```
   [edit policies group group1 list list1 rule rule1 traffic-condition condition1
     protocol-port-condition]
   user@host# set protocol  protocol
   ```

   UDP is the only valid value for PTSP.

3. (Optional) Enter the destination port configuration for the protocol port configuration.

   ```
   [edit policies group group1 list list1 rule rule1 traffic-condition condition1
     protocol-port-condition]
   user@host# edit destination-port
   ```

4. (Optional) Configure the destination port.

   ```
   [edit policies group group1 list list1 rule rule1 traffic-condition condition1
     protocol-port-condition destination-port port]
   user@host# set from-port  from-port
   ```

   Where ***from-port*** is one of the following values:

   - service_port—A predefined global parameter that is the port of the service as specified by the service object
   - Integer in the range 0–65535
   - Expression—A range of port numbers; for example, 10..20
   - Parameter of type port

   Use a range of ports to specify port numbers that are greater than or less than a specified port number. For example:

   - To set a range of ports that is greater than 10, use 11..65535.

- To set a range of ports that is less than 200, use 0..199.

5. (Optional) Enter the source port configuration for the protocol port configuration.

```
user@host# up

[edit policies group group1 list list1 rule rule1 traffic-condition condition1
  protocol-port-condition]
user@host# edit source-port
```

6. (Optional) Configure the source port.

```
[edit policies group group1 list list1 rule rule1 traffic-condition condition1
  protocol-port-condition source-port port]
user@host# set from-port from-port

[edit policies group group1 list list1 rule rule1 traffic-condition condition1
  protocol-port-condition source-port port]
user@host# up
```

Where ***from-port*** is one of the following values:

- service_port — A predefined global parameter that is the port of the service as specified by the service object.
- Integer in the range 0–65535
- Expression — A range of port numbers; for example, 10..20.
- Parameter of type port

Use a range of ports to specify port numbers that are greater than or less than a specified port number. For example:

- To set a range of ports that is greater than 10, use 11..65535.
- To set a range of ports that is less than 200, use 0..199.

7. (Optional) Verify your protocol condition configuration.

```
[edit policies group group1 list list1 rule rule1 traffic-condition
condition1 protocol-port-condition]
user@host# show
protocol udp;
destination-port {
  port {
        from-port service_port;
  }
}
source-port {
  port {
       from-port service_port;
  }
}
```

Configuring Protocol Conditions with Parameters for PTSP Classify-Traffic Conditions (SRC CLI)

Use the following configuration statements to configure classify-traffic conditions that contain a parameter value for the protocol:

```
policies group name list name rule name traffic-condition name
  parameter-protocol-condition {
  protocol  protocol;
}
policies group name list name rule name traffic-condition name
  parameter-protocol-condition proto-attr destination-port port {
  from-port  from-port;
}
policies group name list name rule name traffic-condition name
  parameter-protocol-condition proto-attr source-port port {
  from-port  from-port;
}
```

To configure a protocol condition that contains a parameter value for the protocol:

1. From configuration mode, enter the parameter protocol condition configuration. For example:

   ```
   user@host# edit policies group group1 list list1 rule rule1  traffic-condition condition1
     parameter-protocol-condition
   ```

2. Assign a parameter as the protocol matched by this classify-traffic condition.

   Before you assign a parameter, you must create a parameter of type protocol and commit the parameter configuration.

   ```
   [edit policies group group1 list list1 rule rule1 traffic-condition condition1
     parameter-protocol-condition]
   user@host# set protocol   protocol
   ```

3. (Optional) Enter the protocol attribute configuration.

   ```
   [edit policies group group1 list list1 rule rule1 traffic-condition condition1
     parameter-protocol-condition]
   user@host# edit proto-attr
   ```

4. (Optional) Enter the destination port configuration.

   ```
   [edit policies group group1 list list1 rule rule1 traffic-condition condition1
     parameter-protocol-condition proto-attr]
   user@host# edit destination-port port
   ```

5. (Optional) Configure the TCP or UDP destination port.

   ```
   [edit policies group group1 list list1 rule rule1 traffic-condition condition1
     parameter-protocol-condition proto-attr destination-port port]
   user@host# set from-port  from-port
   ```

   Where ***from-port*** is one of the following values:

- service_port—A predefined global parameter that is the port of the service as specified by the service object.
- Integer in the range 0–65535.
- Expression—A range of port numbers; for example, 10..20.
- Parameter of type port.

Use a range of ports to specify port numbers that are greater than or less than a specified port number. For example:

- To set a range of ports that is greater than 10, use 11..65535.
- To set a range of ports that is less than 200, use 0..199.

6. (Optional) Enter the source port configuration.

```
[edit policies group group1 list list1 rule rule1 traffic-condition condition1
   parameter-protocol-condition proto-attr destination-port port]
user@host# up
[edit policies group group1 list list1 rule rule1 traffic-condition condition1
   parameter-protocol-condition proto-attr]
user@host# edit source-port port
```

7. (Optional) Configure the TCP or UDP source port.

```
[edit policies group group1 list list1 rule rule1 traffic-condition condition1
   parameter-protocol-condition proto-attr source-port port]
user@host# set from-port from-port
```

```
[edit policies group group1 list list1 rule rule1 traffic-condition condition1
   parameter-protocol-condition proto-attr source-port port]
user@host# up
```

```
[edit policies group group1 list list1 rule rule1 traffic-condition condition1
   parameter-protocol-condition proto-attr source-port]
user@host# up
```

```
[edit policies group group1 list list1 rule rule1 traffic-condition condition1
   parameter-protocol-condition proto-attr ]
user@host# up
```

Where ***from-port*** is one of the following values:

- service_port — A predefined global parameter that is the port of the service as specified by the service object.
- Integer in the range 0–65535.
- Expression — A range of port numbers; for example, 10..20.
- Parameter of type port.

Use a range of ports to specify port numbers that are greater than or less than a specified port number. For example:

- To set a range of ports that is greater than 10, use 11..65535.
- To set a range of ports that is less than 200, use 0..199.

8. (Optional) Verify the parameter protocol configuration.

```
[edit policies group group1 list list1 rule rule1 traffic-condition
condition1 parameter-protocol-condition]
user@host# show
protocol protocol;
  destination-port {
    port {
      from-port service_port;
    }
  }
}
```

Configuring TCP Conditions for PTSP Classify-Traffic Conditions (SRC CLI)

Use the following configuration statements to add TCP conditions to a PTSP classify-traffic condition:

policies group *name* list *name* rule *name* traffic-condition *name* tcp-condition {
  protocol *tcp*;
}

Because the protocol is already set to TCP, do not change the protocol or protocol-operation options.

policies group *name* list *name* rule *name* traffic-condition *name* tcp-condition
  destination-port port {
  from-port *from-port*;
}

policies group *name* list *name* rule *name* traffic-condition *name* tcp-condition
  source-port port {
  from-port *from-port*;
}

To add TCP conditions to a PTSP classify-traffic condition:

1. From configuration mode, enter the TCP configuration. For example:

   user@host# **edit policies group group1 list list1 rule rule1 traffic-condition condition1 tcp-condition**

2. (Optional) Enter the protocol for the TCP configuration.

   [edit policies group group1 list list1 rule rule1 traffic-condition condition1 tcp-condition]
   user@host# **set protocol** *protocol*

   For PTSP this is set to TCP.

3. (Optional) Enter the destination port configuration for the TCP configuration.

   [edit policies group group1 list list1 rule rule1 traffic-condition condition1 tcp-condition]
   user@host# **edit destination-port** *port*

4. (Optional) Configure the destination port.

   ```
   [edit policies group group1 list list1 rule rule1 traffic-condition condition1 tcp-condition
     destination-port port]
   user@host# set from-port  from-port
   ```

   Where ***from-port*** is one of the following values:

   - service_port—A predefined global parameter that is the port of the service as specified by the service object.
   - Integer in the range 0–65535.
   - Expression—A range of port numbers; for example, 10..20.
   - Parameter of type port.

   Use a range of ports to specify port numbers that are greater than or less than a specified port number. For example:

   - To set a range of ports that is greater than 10, use 11..65535.
   - To set a range of ports that is less than 200, use 0..199.

5. (Optional) Enter the source port configuration for the TCP configuration.

   ```
   [edit policies group group1 list list1 rule rule1 traffic-condition condition1 tcp-condition
     source-port port]
   user@host# up

   [edit policies group group1 list list1 rule rule1 traffic-condition condition1]
   user@host# edit source-port port
   ```

6. (Optional) Configure the source port.

   ```
   [edit policies group group1 list list1 rule rule1 traffic-condition condition1 tcp-condition
     source-port port]
   user@host# set from-port  from-port

   [edit policies group group1 list list1 rule rule1 traffic-condition condition1 tcp-condition
     source-port port]
   user@host# up
   ```

   Where ***from-port*** is one of the following values:

   - service_port — A predefined global parameter that is the port of the service as specified by the service object.
   - Integer in the range 0–65535
   - Expression — A range of port numbers; for example, 10..20.
   - Parameter of type port

   Use a range of ports to specify port numbers that are greater than or less than a specified port number. For example:

- To set a range of ports that is greater than 10, use 11..65535.
- To set a range of ports that is less than 200, use 0..199.

7. (Optional) Verify the TCP condition configuration.

```
[edit policies group group1 list list1 rule rule1 traffic-condition
condition1 tcp-condition]
user@host# show
protocol tcp;
protocol-operation is;
destination-port {
  port {
        from-port service_port;
  }
}
source-port {
  port {
       from-port service_port;
  }
}
```

Configuring Traffic Match Conditions for PTSP Classify-Traffic Conditions (SRC CLI)

Use the following configuration statements to configure traffic match conditions for PTSP classify traffic conditions.

```
policies group name list name rule name traffic-condition name traffic-match-condition
  {
  term-precedence term-precedence
  application [application...];
  application-group [application-group...];
}
```

To add traffic match conditions to PTSP classify-traffic conditions:

1. From configuration mode, enter the traffic condition configuration. For example:

   user@host# edit policies group group1 list list1 rule rule1 traffic-condition condition1 traffic-match-condition

2. (Optional) Configure the term-precedence for this term in a given policy in relation to other terms. Lower precedence terms are searched first. Precedence matters only within the same class of policies, either dynamic or static. Terms with same precedence may be evaluated in any order.

   [edit policies group group1 list list1 rule rule1 traffic-condition condition1 traffic-match-condition]
   user@host# set term-precedence *term-precedence*

   Enter an integer in the range 1–254.

3. (Optional) Configure the application protocol to match.

   [edit policies group group1 list list1 rule rule1 traffic-condition condition1 traffic-match-condition]
   user@host# set application-protocol *application-protocol*

4. (Optional) Configure a list of application groups to match for this policy.

   [edit policies group group1 list list1 rule rule1 traffic-condition condition1 traffic-match-condition]
   user@host# set application-group *application-group*

5. (Optional) Verify the filter condition configuration.

```
[edit policies group group1 list list1 rule rule1 traffic-condition
condition1 traffic-match-condition]
user@host# show
term–precedence 100;
application–group group1;
}
```

Configuring PTSP Actions

Actions define the action taken on packets that match conditions in a policy rule. You create actions within policy rules.

Topics that discuss how to configure PTSP actions include:

- Configuring Policer-Ref Actions (SRC CLI) on page 209
- Configuring Forwarding Instance Actions (SRC CLI) on page 210
- Configuring Forwarding Class Actions (SRC CLI) on page 211
- Configuring Filter Actions (SRC CLI) on page 212

Configuring Policer-Ref Actions (SRC CLI)

Use this action to specify an action that references a PTSP policer instance. You can configure policer ref actions for PTSP policy rules. The policer instance can be shared by different service rules inside the same policy list. Multiple policy rules can reference the same policer instance, so that the traffic matched by those rules is policed by the same policer instance. If the policer instance is shared, all packets matching any of the service rules are policed together.

NOTE: For PTSP you must:

- Set the role of the policy list to junos-ptsp
- Set the policy list rule type to ptsp-service-rule
- Set the policy list applicability option to both
- Create a policer instance.

Use the following configuration statements to configure a policer-ref action:

```
policies group name list name rule name policer-ref {
  policer-ref policer-name ;
  description description ;
}
```

To configure the policer-ref action:

1. From configuration mode, access the statements for the policer-ref action.

   user@host# edit policies group *name* list *name* rule *name* policer-ref

2. Specify the name of the policer instance you want to reference.

   [edit policies group name list name rule name policer-ref]
   user@host# set *policer–name*

3. Enter a description for the action.

   [edit policies group name list name rule name policer-ref]
   user@host# set description *description*

Configuring Forwarding Instance Actions (SRC CLI)

You can configure forwarding instance actions for routers running the PTSP feature. This action specifies a forwarding instance to assign to flows that match this policy.

NOTE: For PTSP you must:

- Set the role of the policy list to junos-ptsp
- Set the policy list rule type to ptsp-service-rule
- Set the policy list applicability option to both
- Create a policer instance.

Use the following configuration statements to configure a forwarding instance action:

```
policies group name list name rule name forwarding-instance {
  forwarding-instance;
  forwarding-unit forwarding-unit ;
  description description ;
}
```

To configure a forwarding instance action:

1. From configuration mode, enter the forwarding instance configuration. For example:

   user@host# edit policies group group1 list list1 rule rule1 forwarding-instance

2. (Optional) Specify a forwarding instance to assign to flows matching the policy.

   user@host# edit policies group group1 list list1 rule rule1 forwarding-instance “_same_”

   Allowed values are __same__, or one of the forwarding instances configured on the router. The value __same__ forwards the flow in whatever forwarding instance it came in or is set from static configuration.

3. (Optional) Specify a forwarding unit to assign to flows matching this policy. Forwarding unit specifies the multiservice interface unit number for forward flows

to in order to reach the forwarding instance specified by the attribute forwarding-instance. Note that there is only a very loose coupling between this unit number and the forwarding instance. The binding between them only happens with the aid of additional router configuration.

```
[edit policies group group1 list list1 rule rule1 forwarding-instance]
user@host# set forwarding-unit forwarding-unit
```

Enter a value in the range 0–16384.

4. (Optional) Enter a description for the forwarding instance action.

```
[edit policies group group1 list list1 rule rule1 forwarding-instance]
user@host# set description description
```

5. (Optional) Verify the forwarding instance action configuration.

```
[edit policies group bod list input rule pr forwarding-instance]
user@host# show
"'fi1'" forwarding-instance 1 description fi-sample
```

Configuring Forwarding Class Actions (SRC CLI)

You can configure forwarding class actions for JUNOS filter policy rules. The forwarding class action causes the router to assign a forwarding class to packets that match the associated classify-traffic condition.

The type of action that you can create depends on the type of policy rule. See Policy Information Model.

Use the following configuration statements to configure a forwarding class action:

```
policies group name list name rule name forwarding-class {
  forwarding-class;
  description description;
}
```

To configure a forwarding class action:

1. From configuration mode, enter the forwarding class action configuration.

```
user@host# edit policies group bod list input rule pr forwarding-class
```

2. (Optional) Configure the name of the forwarding class assigned to packets.

```
[edit policies group bod list input rule pr forwarding-class]
user@host# set forwarding-class
```

3. (Optional) Enter a description for the forwarding class action.

```
[edit policies group bod list input rule pr forwarding-class]
user@host# set description description
```

4. (Optional) Verify the forwarding class action configuration.

```
[edit policies group bod list input rule pr forwarding-class]
user@host# show
forwarding-class fc_expedited;
description "Expedited forwarding class";
```

Configuring Filter Actions (SRC CLI)

Use this action to discard packets. You can configure filter actions for JUNOS filters and JUNOSe policy rules. The type of action that you can create depends on the type of policy rule. See Policy Information Model.

Use the following configuration statement to configure a filter action:

```
policies group name list name rule name filter {
  description description ;
}
```

To configure a filter action:

1. From configuration mode, enter the filter action configuration.

   user@host# edit policies group junos_filter list in rule pr filter

2. (Optional) Enter a description for the filter action.

   [edit policies group junos_filter list in rule pr filter]
   user@host# set description *description*

3. (Optional) Verify the filter action configuration.

   ```
   [edit policies group junos_filter list in rule pr filter]
   user@host# show
   description "Filter action for JUNOS policies";
   ```

Example: Configuring the SRC Software to Support PTSP on the MX Series Router

The following example illustrates how to configure two SAEs running on hosts *src1* and *src2* to manage packet-triggered subscriber sessions on all routing instances of an MX Series router:

```
shared {
  network {
    device mx-name {
      origin-host mx-origin-host;
      device-type junos-ptsp;
      peers mx-name-peer;
      virtual-router * {
        sae-connection [src1 src2];
      }
    }
  }
  diameter {
    peer mx-name-peer {
      address 10.0.0.1;
```

```
        port 3868;
        protocol tcp;
      }
    }
  }
  policies group MXPolicy list ptsp {
    role junos-ptsp;
    applicability both;
    policer MXpolicer {
      bandwidth 100000;
      max-burst-size 10000;
    }
    rule ptsp-r1 {
      type service-rule;
      precedence 100;
      accounting;
      traffic-condition 1 {
        destination-network {
          group-network network-specifier 1.2.3.0/24;
        }
        match-direction both;
      }
      traffic-condition 2 {
        destination-network {
          group-network network-specifier 2.3.4.0/23;
        }
        match-direction input;
      }
      forwarding-instance 1 forwarding-unit 2;
    }
    rule ptsp-r2 {
      type service-rule;
      precedence 100;
      traffic-condition 1 {
        destination-network {
          group-network network-specifier 3.4.5.0/24;
        }
        match-direction both;
      }
      filter;
    }
  }
  shared {
    sae {
      group <name-of-config-group> {
        configuration {
          driver {
            junos-ptsp {
              cached-driver-expiration 600;
              keep-alive-timeout 60;
              registry-retry-interval 30;
              reply-timeout 20;
              sae-community-manager PTSPCommunityManager;
              sequential-message-timeout 20;
              session-store {
                communications-timeout 60000;
```

```
                disk-load-buffer-size 1000000;
                idle-timeout 3600000;
                load-timeout 420000;
                maximum-backlog-ratio 1.5;
                maximum-file-size 25000000;
                maximum-queue-age 100;
                maximum-queue-size 51050;
                maximum-queued-operations 50;
                maximum-session-size 10000;
                minimum-backlog 5000000;
                minimum-disk-space-usage 25;
                network-buffer-size 51050;
                retry-interval 300000;
                rotation-batch-size 50;
            }
            thread-idle-timeout 60;
            thread-pool-size 50;
        }
    }
  }
 }
}
```

The active SAE registers events from the MX Series router based on the configured origin host (mx-origin-host in the example). If the origin host is not configured, the SAE uses the device name (mx-name) instead.

Example: Configuring the SRC Software to Support Both PTSP and JSRC on the MX Series Router

If you are using both the *junos-ise* device driver for the JSRC feature and the PTSP device driver, you need to configure two network device entries, one for each device driver. For example:

```
shared {
  network {
    device mx-name-ptsp {
      origin-host mx-origin-host;
      device-type junos-ptsp;
      peers mx-name-peer;
      virtual-router * {
        sae-connection [src1 src2];
      }
    }
    device mx-name-jsrc {
      origin-host mx-origin-host;
      device-type junos-ise;
      peers mx-name-peer;
      virtual-router * {
        sae-connection [src2 src3];
      }
    }
  }
  diameter {
    peer mx-name-peer {
```

```
        address 10.0.0.1;
        port 3868;
        protocol tcp;
      }
    }
  }
```

In this example, the JSRC and PTSP device drivers are being used simultaneously. Notice:

- The two device entries have different names (mx-name-ptsp and mx-name-jsrc).
- Both the device entries contain the origin host attribute that matches the diameter host as configured in the MX Series router .

If the origin host is not specified, the name of the device is used instead. In other words, it is also possible to configure two entries, where the name of one entry matches the Diameter host name of the MX Series router (without origin host), and the second entry contains the origin host. For example:

```
shared network {
  device mx-origin-host { ... }
  device mx-origin-host-2 {
    origin-host mx-origin-host;
    ...
  }
}
```

CHAPTER 19

Managing Subscriber Sessions on MX Series Routers in an SRC Network

Overview of Subscriber Sessions on MX Series Routers

The SRC software can manage subscriber sessions on MX Series routers. Common types of subscriber sessions on MX Series routers include:

- One interface subscriber session for each statically configured VLAN.
- One address subscriber session for each DHCP address.

You can manage subscriber sessions with the External Subscriber Monitor application and the CoA script service. You can use External Subscriber Monitor to authorize access requests from the MX Series router and to log in or log out authorized subscribers. You can use the pseudo-RADIUS authorization server in External Subscriber Monitor to limit the number of DHCP leases for a subscriber by specifying the interface-name attribute in the subscriber profile and then setting a parameter substitution for the dhcpLeaseLimit parameter for that interface. You can configure the CoA script service to dynamically activate or deactivate services on the MX Series router. This method uses RADIUS attributes and RADIUS vendor-specific attributes (VSAs) to identify a subscriber session whose services are to be activated or deactivated.

Managing Subscriber Sessions on MX Series Routers (SRC CLI)

The following topics provide procedures that allow you to manage subscriber sessions on MX Series routers with the SRC CLI:

Configuring External Subscriber Monitor (SRC CLI)

Use External Subscriber Monitor to log in and log out authorized subscribers and to provide interim updates for authorized subscribers.

To configure External Subscriber Monitor as a pseudo–RADIUS accounting server:

1. From configuration mode, access the configuration statement that configures the local properties.

   ```
   user@host# edit slot 0 external-subscriber-monitor
   ```

2. Configure the local properties for External Subscriber Monitor.

   If you are configuring the pseudo–RADIUS authorization server, specify the **include-mac-address** and **include-interface-name** options when configuring External Subscriber Monitor so that the MAC address and interface name attributes are included in the event notifications sent to the SAE.

   ```
   [edit slot 0 external-subscriber-monitor]
   user@host# set ?
   ```

   For more information about configuring External Subscriber Monitor, see Configuring External Subscriber Monitor (SRC CLI).

Configuring Pseudo–RADIUS Authorization Server Properties (SRC CLI)

Tasks to configure the pseudo–RADIUS authorization server are:

Configuring the Pseudo–RADIUS Authorization Server (SRC CLI)

Use the following configuration statements to configure the pseudo–RADIUS authorization server:

```
slot number external-subscriber-monitor radius-authorization {
    port port;
    local-address local-address;
    check-lease-limit-with-sae;
    query-cached-dhcp-profile;
    default-lease-limit default-lease-limit;
```

```
    invalid-pool-name invalid-pool-name;
    lease-time-limit lease-time-limit;
    cleanup-interval cleanup-interval;
    maximum-age maximum-age;
    minimum-pool-size minimum-pool-size;
    maximum-queue-length maximum-queue-length;
    service-type (all | login | framed | callback-login | callback-framed | outbound |
      administrative | nas-prompt | authenticate-only | callback-nas-prompt | callback-check
      | callback-administrative);
  }
  slot number external-subscriber-monitor radius-authorization client client-address {
    secret secret;
  }
```

To configure the pseudo–RADIUS authorization server:

1. From configuration mode, access the configuration statement that configures the pseudo–RADIUS authorization server.

   ```
   user@host# edit slot 0 external-subscriber-monitor radius-authorization
   ```

2. Specify the listening port for RADIUS requests.

   ```
   [edit slot 0 external-subscriber-monitor radius-authorization]
   user@host# set port port
   ```

3. (Optional) Specify the host address to bind to the pseudo–RADIUS authorization server. Absence (or deletion) of this attribute means binding it to a wildcard (*) address.

   ```
   [edit slot 0 external-subscriber-monitor radius-authorization]
   user@host# set local-address local-address
   ```

4. (Optional) Specify whether to query the SAE for the number of active subscribers for a given interface. If set to true, the response to the RADIUS access request depends on the comparison between the number of active subscriber sessions and the lease limit for the interface. If the number of active subscriber sessions is less than the lease limit, the response is the RADIUS access accept message without the lease limit RADIUS attribute; otherwise, the response is the RADIUS access accept message where the subscriber is not assigned an address. If set to false, the response is the RADIUS access accept message with the lease limit RADIUS attribute. If the lease limit RADIUS vendor-specific attribute is returned, the MX Series router verifies the lease limit.

   ```
   [edit slot 0 external-subscriber-monitor radius-authorization]
   user@host# set check-lease-limit-with-sae
   ```

5. (Optional) Specify whether to search for a cached DHCP profile in the o=AuthCache directory based on the MAC address. If set to true, you must configure a directory connection to the cached DHCP profiles. See Configuring Directory Connection Properties for the Cached DHCP Profiles on page 121.

   If set to true, the following conditions apply:

   - If a cached DHCP profile is found, the RADIUS response message includes the RADIUS attribute values for framed IP address, pool name, service bundle, and RADIUS class attributes that are present in the cached DHCP profile.

- If the **check-lease-limit-with-sae** option is set to true and the number of active subscriber sessions is less than the lease limit, the RADIUS access accept message includes the cached DHCP profile.
- If the **check-lease-limit-with-sae** option is set to false, the RADIUS response includes the lease limit.

If set to false, the RADIUS response message does not include the cached DHCP profile information.

```
[edit slot 0 external-subscriber-monitor radius-authorization]
user@host# set query-cached-dhcp-profile
```

6. (Optional) Specify the default lease limit for all interfaces.

```
[edit slot 0 external-subscriber-monitor radius-authorization]
user@host# set default-lease-limit default-lease-limit
```

7. Specify the invalid pool name returned when the number of active subscriber sessions exceeds the lease limit.

```
[edit slot 0 external-subscriber-monitor radius-authorization]
user@host# set invalid-pool-name invalid-pool-name
```

8. (Optional) Specify the timeout of a cached authenticated request.

```
[edit slot 0 external-subscriber-monitor radius-authorization]
user@host# set lease-time-limit lease-time-limit
```

9. Specify the amount of time to wait before cleaning up cached RADIUS access requests that have been accepted.

```
[edit slot 0 external-subscriber-monitor radius-authorization]
user@host# set cleanup-interval cleanup-interval
```

10. Specify the maximum age of an unacknowledged RADIUS access request cached in memory. We recommend a value slightly greater than the RADIUS packets retry interval.

```
[edit slot 0 external-subscriber-monitor radius-authorization]
user@host# set maximum-age maximum-age
```

11. Specify the minimum number of concurrent threads processing RADIUS access messages subtasks.

```
[edit slot 0 external-subscriber-monitor radius-authorization]
user@host# set minimum-pool-size minimum-pool-size
```

12. Specify the maximum number of unacknowledged RADIUS messages to be received from the RADIUS server before it discards new messages.

```
[edit slot 0 external-subscriber-monitor radius-authorization]
user@host# set maximum-queue-length maximum-queue-length
```

13. Specify the service type of the RADIUS packets that will be forwarded.

```
[edit slot 0 external-subscriber-monitor radius-authorization]
user@host# set service-type service-type
```

14. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor radius-authorization]
```

```
user@host# show
```

15. Access the configuration statement that specifies the trusted RADIUS clients.

```
[edit slot 0 external-subscriber-monitor radius-authorization]
user@host# edit client client-address
[edit slot 0 external-subscriber-monitor radius-authorization client client-address]
```

16. Specify the RADIUS shared secret for the client.

```
[edit slot 0 external-subscriber-monitor radius-authorization client client-address]
user@host# set secret secret
```

Configuring the Directory Connection Properties for the Subscriber Data

The subscriber data can be queried for information such as the interface's lease limit.

Use the following statements to configure the directory connection to the directory in which the subscriber data is stored:

```
slot number external-subscriber-monitor radius-authorization ldap subscriber-data {
    base base;
    base-dn base-dn;
}
slot number external-subscriber-monitor radius-authorization ldap subscriber-data
    directory-connection {
    url url;
    principal principal;
    credentials credentials;
    protocol (ldaps);
    backup-urls [backup-urls...];
    timeout timeout;
    check-interval check-interval;
    blacklist;
    snmp-agent;
    signature-dn signature-dn;
}
```

To configure directory connection properties:

1. From configuration mode, access the configuration statement that configures the directory connection.

```
user@host# edit slot 0 external-subscriber-monitor radius-authorization ldap
  subscriber-data
```

2. Specify the top-level directory DN.

```
[edit slot 0 external-subscriber-monitor radius-authorization ldap subscriber-data]
user@host# set base base
```

3. Specify the subtree in the directory in which the subscriber data is stored.

```
[edit slot 0 external-subscriber-monitor radius-authorization ldap subscriber-data]
user@host# set base-dn base-dn
```

4. Access the configuration statement that configures the directory connection properties.

```
[edit slot 0 external-subscriber-monitor radius-authorization ldap subscriber-data]
user@host# edit directory-connection
```

5. Specify the directory connection properties for the subscriber data.

```
[edit slot 0 external-subscriber-monitor radius-authorization ldap subscriber-data
  directory-connection]
user@host# set ?
```

6. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor radius-authorization ldap subscriber-data]
user@host# show
```

Configuring Directory Connection Properties for the Cached DHCP Profiles

The DHCP profiles can be queried by MAC address for the RADIUS framed IP address for authorized subscribers or invalid pool name for unauthorized subscribers.

Use the following statements to configure the directory connection to the directory in which the cached DHCP profiles are stored:

```
slot number external-subscriber-monitor radius-authorization ldap cached-dhcp-profile
  {
  base base;
  base-dn base-dn;
}
slot number external-subscriber-monitor radius-authorization ldap cached-dhcp-profile
  directory-connection {
  url url;
  principal principal;
  credentials credentials;
  protocol (ldaps);
  backup-urls [backup-urls...];
  timeout timeout;
  check-interval check-interval;
  blacklist;
  snmp-agent;
  signature-dn signature-dn;
}
```

To configure directory connection properties:

1. From configuration mode, access the configuration statement that configures the directory connection.

```
user@host# edit slot 0 external-subscriber-monitor radius-authorization ldap
  cached-dhcp-profile
```

2. Specify the top-level directory DN.

```
[edit slot 0 external-subscriber-monitor radius-authorization ldap cached-dhcp-profile]
user@host# set base base
```

3. Specify the subtree in the directory in which the cached DHCP profiles are stored.

```
[edit slot 0 external-subscriber-monitor radius-authorization ldap cached-dhcp-profile]
user@host# set base-dn base-dn
```

4. Access the configuration statement that configures the directory connection properties.

```
[edit slot 0 external-subscriber-monitor radius-authorization ldap cached-dhcp-profile]
```

```
user@host# edit directory-connection
```

5. Specify the directory connection properties for the cached DHCP profiles.

```
[edit slot 0 external-subscriber-monitor radius-authorization ldap cached-dhcp-profile
  directory-connection]
user@host# set ?
```

6. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor radius-authorization ldap cached-dhcp-profile]
user@host# show
```

Configuring the NIC Proxy for the Pseudo-RADIUS Authorization Server (SRC CLI)

When the **check-lease-limit-with-sae** option is set to true, you must configure the NIC proxy so that the pseudo-RADIUS authorization server can find the SAE managing the interface and determine the number of subscriber sessions already established on the interface (that is, the number of leases on the interface). The NIC proxy must be configured for a NIC scenario that maps VRs to SAEs.

Tasks to configure the NIC proxy are:

- Configuring Resolution Information for a NIC Proxy on page 223
- Changing the Configuration for the NIC Proxy Cache on page 224
- Configuring a NIC Proxy for NIC Replication on page 224

Configuring Resolution Information for a NIC Proxy

Use the following configuration statements to configure the NIC proxy:

```
slot number external-subscriber-monitor nic-proxy-configuration radius-authorization-nic
  resolution {
  resolver-name resolver-name;
  constraints constraints;
}
```

To configure resolution information for a NIC proxy:

1. From configuration mode, access the configuration statement that configures the NIC proxy configuration. In this sample procedure, the NIC proxy called radius-authorization-nic is configured.

```
user@host# edit slot 0 external-subscriber-monitor nic-proxy-configuration
  radius-authorization-nic resolution
```

2. Specify the resolution information for this NIC proxy.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration
  radius-authorization-nic resolution]
user@host# set ?
```

   For more information about configuring resolution information for a NIC proxy, see Configuring Resolution Information for a NIC Proxy (SRC CLI).

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration
  radius-authorization-nic resolution]
```

```
user@host# show
```

Changing the Configuration for the NIC Proxy Cache

You can modify cache properties for the NIC proxy to optimize the resolution performance for your network configuration and system resources. Typically, you can use the default settings for the cache properties. The configuration statements are available at the Advanced editing level.

Use the following configuration statements to change values for the NIC proxy cache:

```
slot number external-subscriber-monitor nic-proxy-configuration radius-authorization-nic
  cache {
  cache-size cache-size;
  cache-cleanup-interval cache-cleanup-interval;
  cache-entry-age cache-entry-age;
}
```

To configure the cache for a NIC proxy:

1. From configuration mode, access the configuration statement that specifies the NIC proxy configuration. In this sample procedure, the NIC proxy called radius-authorization-nic is configured.

   ```
   user@host# edit slot 0 external-subscriber-monitor nic-proxy-configuration
     radius-authorization-nic cache
   ```

2. Specify the cache properties for the NIC proxy.

   ```
   [edit slot 0 external-subscriber-monitor nic-proxy-configuration
     radius-authorization-nic cache]
   user@host# set ?
   ```

   For more information about configuring the cache for a NIC proxy, see Changing the Configuration for the NIC Proxy Cache (SRC CLI).

3. (Optional) Verify your configuration.

   ```
   [edit slot 0 external-subscriber-monitor nic-proxy-configuration
   radius-authorization-nic cache]
   user@host# show
   cache-size 10000;
   cache-cleanup-interval 15;
   ```

Configuring a NIC Proxy for NIC Replication

Typically, you configure NIC replication to keep the NIC highly available. You configure NIC host selection to specify the groups of NIC hosts to be contacted to resolve a request, and to define how the NIC proxy handles NIC hosts that the proxy is unable to contact. The configuration statements are available at the Advanced editing level.

Use the following configuration statements to configure NIC host selection for a NIC proxy:

```
slot number external-subscriber-monitor nic-proxy-configuration radius-authorization-nic
  nic-host-selection {
  groups groups;
  selection-criteria (roundRobin | randomPick | priorityList);
}
```

```
slot number external-subscriber-monitor nic-proxy-configuration radius-authorization-nic
  nic-host-selection blacklisting {
  try-next-system-on-error;
  number-of-retries-before-blacklisting number-of-retries-before-blacklisting;
  blacklist-retry-interval blacklist-retry-interval;
}
```

To configure a NIC proxy to use NIC replication:

1. From configuration mode, access the configuration statement that specifies the NIC proxy configuration. In this sample procedure, the NIC proxy called radius-authorization-nic is configured.

   ```
   user@host# edit slot 0 external-subscriber-monitor nic-proxy-configuration
     radius-authorization-nic nic-host-selection
   ```

2. (Optional) Configure NIC host selection for a NIC proxy.

   ```
   [edit slot 0 external-subscriber-monitor nic-proxy-configuration
     radius-authorization-nic nic-host-selection]
   user@host# set ?
   ```

   For more information about configuring NIC host selection for a NIC proxy, see Configuring a NIC Proxy for NIC Replication (SRC CLI).

3. (Optional) Verify your configuration.

   ```
   [edit slot 0 external-subscriber-monitor nic-proxy-configuration
   radius-authorization-nic nic-host-selection]
   user@host# show
   groups ;
   selection-criteria roundRobin;
   ```

4. Access the configuration statement that specifies the NIC proxy configuration for blacklisting—the process of handling nonresponsive NIC hosts.

   ```
   [edit slot 0 external-subscriber-monitor nic-proxy-configuration
     radius-authorization-nic nic-host-selection]
   user@host# edit blacklisting
   [edit slot 0 external-subscriber-monitor nic-proxy-configuration
     radius-authorization-nic nic-host-selection blacklisting]
   ```

5. (Optional) Configure blacklisting for a NIC proxy.

   ```
   [edit slot 0 external-subscriber-monitor nic-proxy-configuration
     radius-authorization-nic nic-host-selection blacklisting]
   user@host# set ?
   ```

   For more information about configuring NIC host selection for a NIC proxy, see Configuring a NIC Proxy for NIC Replication (SRC CLI).

6. (Optional) Verify your configuration.

   ```
   [edit slot 0 external-subscriber-monitor nic-proxy-configuration
   radius-authorization-nic nic-host-selection blacklisting]

   user@host# show

   [edit slot 0 external-subscriber-monitor nic-proxy-configuration
   radius-authorization-nic nic-host-selection blacklisting]
   user@host# show
   try-next-system-on-error;
   ```

```
number-of-retries-before-blacklisting 3;
blacklist-retry-interval 15;
```

Extracting RADIUS Attributes with the Pseudo–RADIUS Authorization Server (SRC CLI)

The pseudo–RADIUS authorization server extracts RADIUS attribute values from the MX Series router for which it receives access requests.

Tasks to configure the RADIUS attribute value extraction are:

- Extracting Interface Name Attribute Values on page 226
- Extracting Virtual Router Name Attribute Values on page 226

Extracting Interface Name Attribute Values

The interface name value is the subscriber line interface. This value is extracted from the NAS-Port-ID attribute. The default settings for this configuration are sufficient for most applications.

Use the following configuration statements to extract the interface name value from the RADIUS access request:

```
slot number external-subscriber-monitor radius-attribute-extraction default interface-name
  {
  regular-expression [regular-expression...];
}
```

To extract the interface name value:

1. From configuration mode, access the configuration statement that configures RADIUS attribute extraction for the interface name value.

   ```
   user@host# edit slot 0 external-subscriber-monitor radius-attribute-extraction
     default interface-name
   ```

2. (Optional) Specify the RADIUS attribute value format with a regular expression. You can group regular expressions by enclosing them in parentheses. The value for the interface is the part of the NAS-Port-ID matched by the first group in your regular expression. For more information about using regular expressions, see http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html.

   ```
   [edit slot 0 external-subscriber-monitor radius-attribute-extraction default
     interface-name]
   user@host# set regular-expression [regular-expression...]
   ```

   For example, to specify that the extracted interface name value is ge-0/0/3.0 from the NAS-Port attribute value of ge-0/0/3.0[:0-0]:

   ```
   [edit slot 0 external-subscriber-monitor radius-attribute-extraction default
     interface-name]
   user@host# set regular-expression ([a-zA-Z0-9-/.]+)\\[:.*
   ```

Extracting Virtual Router Name Attribute Values

In most cases, the virtual router name value is in the format default@<NAS-ID attribute>. The default settings extract a virtual router name in this format. If your environment is different, you can configure a different format for the extracted value.

Use the following configuration statements to extract the virtual router name value from the RADIUS access request:

```
slot number external-subscriber-monitor radius-attribute-extraction default
  virtual-router-name {
  id id;
  vsa;
  vsa-id vsa-id;
  regular-expression [regular-expression...];
  type (raw-byte | chars);
  prefix prefix;
}
```

To extract the virtual router name value:

1. From configuration mode, access the configuration statement that configures RADIUS attribute extraction for the virtual router name value.

   ```
   user@host# edit slot 0 external-subscriber-monitor radius-attribute-extraction
     default virtual-router-name
   ```

2. Specify the RADIUS attribute identifier.

   ```
   [edit slot 0 external-subscriber-monitor radius-attribute-extraction default
     virtual-router-name]
   user@host# set id id
   ```

3. (Optional) Specify whether the RADIUS attribute is a vendor-specific attribute.

   ```
   [edit slot 0 external-subscriber-monitor radius-attribute-extraction default
     virtual-router-name]
   user@host# set vsa
   ```

4. (Optional) Specify the RADIUS vendor-specific attribute identifier.

   ```
   [edit slot 0 external-subscriber-monitor radius-attribute-extraction default
     virtual-router-name]
   user@host# set vsa-id vsa-id
   ```

5. (Optional) Specify the RADIUS attribute value format with a regular expression. You can group regular expressions by enclosing them in parentheses. The value for the interface is the part of the NAS-Port-ID matched by the first group in your regular expression. For more information about using regular expressions, see http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html.

   ```
   [edit slot 0 external-subscriber-monitor radius-attribute-extraction default
     virtual-router-name]
   user@host# set regular-expression [regular-expression...]
   ```

   For example:

   ```
   [edit slot 0 external-subscriber-monitor radius-attribute-extraction default
     virtual-router-name]
   user@host# set regular-expression ([a-zA-Z0-9-/.]+)\\[:.*
   ```

6. (Optional) Specify the value type of this RADIUS attribute.

   ```
   [edit slot 0 external-subscriber-monitor radius-attribute-extraction default
     virtual-router-name]
   user@host# set type (raw-byte | chars)
   ```

where:

- **raw-byte**—Raw bytes
- **chars**—Sequence of characters

7. (Optional) Specify the prefix that is prepended to the extracted RADIUS attribute value.

```
[edit slot 0 external-subscriber-monitor radius-attribute-extraction default
  virtual-router-name]
user@host# set prefix prefix
```

Enabling the Pseudo-RADIUS Authorization Server (SRC CLI)

To enable the pseudo–RADIUS authorization server, configure the pseudo-RADIUS authorization server and make sure the External Subscriber Monitor is running.

To start External Subscriber Monitor:

```
user@host> enable component extsubmon
```

Disabling the Pseudo-RADIUS Authorization Server (SRC CLI)

To disable the pseudo–RADIUS authorization server, delete the pseudo–RADIUS authorization server configuration for External Subscriber Monitor from configuration mode.

```
[edit slot 0 external-subscriber-monitor]
user@host# delete radius-authorization
```

Setting Up MX Series Routers in the SRC Network (SRC CLI)

To set up the MX Series router so that the router can be managed by the SAE:

1. From configuration mode, access the configuration statement that configures network devices. This sample procedure uses mx_device as the name of the router.

```
user@host# edit slot 0 shared network device mx_device
```

2. Set the type of device to third-party.

```
[edit shared network device mx_device]
user@host# set device-type third-party
```

3. From configuration mode, access the configuration statements for virtual routers. For MX Series routers, use the name default for the virtual router.

```
[edit shared network device mx_device]
user@host# edit virtual-router default
```

4. Specify the addresses of SAEs that can manage this router.

```
[edit shared network device mx_device virtual-router default]
user@host# set sae-connection [sae-connection...]
```

Configuring the CoA Script Service for MX Series Routers (SRC CLI)

To configure the script service for the MX Series router:

1. Create a script service in the services global service name hierarchy or the services scope name service name hierarchy. For example:

   ```
   [edit]
   user@host# edit services global service cos-service
   ```

2. Set the type to script.

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

3. (Optional) Configure other properties as needed for your service.

4. Configure the script properties.

   a. Access the script hierarchy for the configured script service.

      ```
      [edit services global service cos-service]
      user@host# edit script
      ```

   b. Specify URL as the script type.

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

   c. Specify the name of the Java class that implements the script service.

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

   d. Configure the URL of the script service or the path and filename of the service.

      ```
      [edit services global service cos-service script]
      user@host# set file file:///opt/UMC/sae/lib/coa.jar
      ```

      If you specify a file URL, you must copy the file to the C Series Controller. If you specify an ftp or http URL, the file can reside on a centralized server. You can find the *coa.jar* file in the application and SDK distribution on the Juniper Networks Web site at:

      https://www.juniper.net/support/csc/swdist-erx/src.html

      in the *SDK+AppSupport+Demos+Samples.tar.gz* archive file with the pathname:

      AppSupport+Demos+Samples/SDK/scriptServices/coa/lib/coa.jar

5. Verify the configuration.

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

```
}
```

6. Configure the parameters for the script service.

Configuring Parameters for the Script Service for MX Series Routers (SRC CLI)

Provide parameter substitutions with the values that are in the service definitions for the script service.

Table 9 on page 230 lists the parameters specified by the sample script service.

Table 9: Parameter Substitutions for MX Series Routers CoA Services

Parameter Name	Description
dynClientIp	IP address of the device.
dynClientPort	UDP port number of the device.
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 RADIUS packets 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
	  ◦ VSA in the format vendor-specific.<vendor#>.<vsa#>.string
	• pluginEventAttribute—Valid Python expression
	• \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.4874.10.string=’video’\nstop.vendor-specific.4874.10.string=’default’\n"

To configure substitutions for the script parameters:

1. At the hierarchy for the script service, specify substitutions for the parameters. For example:

```
[edit services global service cos-service]
user@host# set parameter substitution [ dynSecret=\"secret\" dynRetry=2
  dynClientIp=10.227.7.111 dynClientPort=9099
  "dynConfig=\"start-stop.1.string=primaryUserName\\nstart-stop.Acct-Session-id=ifSessionId
  \\nstart.vendor-specific.4874.108.string=['T01 3m', 'T04
  consumer-scheduler-map']\\nstop.vendor-specific.4874.108.string=['T01 1m',
  'T04 data-scheduler-map']\\nstart.vendor-specific.4874.10.string='video'
  \\nstop.vendor-specific.4874.10.string='default'\\n\"" ]
```

2. Verify the configuration.

```
[edit services global service cos-service]
user@host# show
```

Configuring Subscriptions to the Script Service

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

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

Viewing Statistics for the Pseudo–RADIUS Authorization Server (SRC CLI)

Purpose View RADIUS statistics for the pseudo–RADIUS authorization server.

Action To display client statistics for the pseudo–RADIUS authorization server:

```
user@host> show external-subscriber-monitor statistics radius-authorization
Client Statistics
Client Address                                    10.227.7.45
Number of received radius access-request          602524
Number of dropped radius access-request           0
Number of radius access-accept sent               602524
Number of radius access-reject sent               0
Number of dropped radius authentication response  0
Number of access request received per second      58
```

To display specific client statistics for the pseudo–RADIUS authorization server:

```
user@host> show external-subscriber-monitor statistics radius-authorization
client-address client-address
```

Monitoring Statistics for the Pseudo–RADIUS Authorization Server (SRC CLI)

Purpose Display real-time RADIUS authorization statistics for the pseudo–RADIUS authorization server.

Action To display real-time client statistics for the pseudo–RADIUS authorization server:

```
user@host> monitor external-subscriber-monitor statistics radius-authorization
client-address client-address
```

PART 4

Index

Index

Q

R

S

T

W