



---

# SRC PE Software

## Solutions Guide

Release

4.13.x



---

Modified: 2019-08-21

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

Copyright © 2019 Juniper Networks, Inc. All rights reserved.

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

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

#### *SRC PE Software Solutions Guide*

Release 4.13.x

Copyright © 2019 Juniper Networks, Inc. All rights reserved.

#### Revision History

August 2019—Revision 1

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

#### YEAR 2000 NOTICE

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

#### 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](http://www.juniper.net/techpubs).

#### END USER LICENSE AGREEMENT

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

# Abbreviated Table of Contents

	About the Documentation .....	xv
Part 1	Providing Specialized Services in an SRC Environment	
Chapter 1	Managing Tiered and Premium Services with QoS on JunosE Routers (SRC CLI) .....	3
Chapter 2	Managing Subscribers for a Wireless Roaming Environment .....	17
Chapter 3	Configuring VoIP Services in an SRC Network .....	25
Chapter 4	Providing Packet Mirroring in an SRC Network .....	29
Part 2	Managing Services in a PCMM Environment	
Chapter 5	Providing Premium Services in a PCMM Environment .....	41
Chapter 6	Configuring the SAE for a PCMM Environment (SRC CLI) .....	57
Chapter 7	Adding Objects for CMTS Devices (SRC CLI) .....	71
Chapter 8	Using the NIC Resolver in a PCMM Environment .....	75
Part 3	Managing Services on RADIUS and Diameter Devices	
Chapter 9	Managing Services on Third-Party Devices in the SRC Network .....	79
Chapter 10	Managing the SRC Diameter Server .....	87
Chapter 11	Monitoring the SRC Diameter Server (SRC CLI) .....	97
Chapter 12	Managing Services with Diameter on MX Series Routers .....	101
Chapter 13	Managing Subscriber Sessions on MX Series Routers in an SRC Network .....	123
Chapter 14	Configuring Services for SRC-Managed Routers .....	143
Chapter 15	Configuring PCC or ePCC Rules for Router Running Junos OS and Acting as PCEF .....	159
Part 4	Using SRC Configuration Wizards	
Chapter 16	SRC Configuration Wizards Overview (SRC CLI) .....	185
Chapter 17	SRC Configuration Wizards Overview (C-Web Interface) .....	189
Chapter 18	Using the Fair Usage on MX Series Routers Configuration Wizard .....	193



# Table of Contents

	<b>About the Documentation</b> . . . . .	<b>xv</b>
	SRC Documentation and Release Notes . . . . .	xv
	Audience . . . . .	xv
	Documentation Conventions . . . . .	xv
	Documentation Feedback . . . . .	xvii
	Requesting Technical Support . . . . .	xviii
	Self-Help Online Tools and Resources . . . . .	xviii
	Creating a Service Request with JTAC . . . . .	xviii
<b>Part 1</b>	<b>Providing Specialized Services in an SRC Environment</b>	
<b>Chapter 1</b>	<b>Managing Tiered and Premium Services with QoS on JunosE Routers (SRC CLI)</b> . . . . .	<b>3</b>
	QoS on JunosE Routers Overview . . . . .	3
	Dynamically Managing QoS Profiles . . . . .	4
	How QoS Profile Tracking Works . . . . .	4
	Identifying QoS Services . . . . .	4
	Determining the QoS Profile . . . . .	5
	Setting Up Policy Groups . . . . .	6
	Setting Up Services . . . . .	7
	Reestablishing Default QoS Profile . . . . .	7
	Example: How QTP Activates a QoS Service . . . . .	7
	Configuring QoS Profile-Tracking Plug-Ins (SRC CLI) . . . . .	9
	Configuring Search Filters for QoS Profile-Tracking Plug-Ins . . . . .	11
	Updating QoS Profile Data in the Directory . . . . .	13
	Query Fields . . . . .	13
	Examples: Searching for QoS Information . . . . .	14
<b>Chapter 2</b>	<b>Managing Subscribers for a Wireless Roaming Environment</b> . . . . .	<b>17</b>
	Wireless Roaming Environment Overview . . . . .	17
	Subscriber Access in a Wireless Roaming Environment . . . . .	17
	Configuring Subscriber Access for a Wireless Location . . . . .	18
	Configuring RADIUS Authentication . . . . .	18
	Creating Subscriber Access to an ISP . . . . .	21
	Creating Web Access . . . . .	22
	Setting Idle Timeout Options for the SAE . . . . .	23

<b>Chapter 3</b>	<b>Configuring VoIP Services in an SRC Network . . . . .</b>	<b>25</b>
	Session Management for VoIP Services Overview . . . . .	25
	Accounting and Tracking . . . . .	25
	VoIP Call Setup . . . . .	26
	Configuring Policies and Services for VoIP . . . . .	26
	Activating VoIP Services for Assigned IP Subscribers . . . . .	27
	Setting Timeouts for Assigned IP Subscriber Sessions . . . . .	28
<b>Chapter 4</b>	<b>Providing Packet Mirroring in an SRC Network . . . . .</b>	<b>29</b>
	Packet-Mirroring Services Overview . . . . .	29
	Configuring Packet-Mirroring Support in an SRC Network . . . . .	30
	Configuring the Script Service for Packet Mirroring . . . . .	31
	Configuring Parameters for the Script Service for Packet Mirroring . . . . .	32
	Specifying Maximum Number of RADIUS Peers (SRC CLI) . . . . .	34
	Example: Using the Sample Packet-Mirroring Application . . . . .	35
	Example: Packet Mirroring for PPP Subscribers . . . . .	36
	Example: Packet Mirroring for DHCP Subscribers . . . . .	36
	Configuring DHCP Subscriber Sessions . . . . .	36
	Disabling RADIUS Authentication for DHCP Subscribers . . . . .	36
	Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API . . . . .	37
<b>Part 2</b>	<b>Managing Services in a PCMM Environment</b>	
<b>Chapter 5</b>	<b>Providing Premium Services in a PCMM Environment . . . . .</b>	<b>41</b>
	PCMM Environment Overview . . . . .	41
	PCMM Architecture . . . . .	41
	DOCSIS Protocol . . . . .	42
	Service Flows . . . . .	43
	Client Types . . . . .	43
	SRC Software in the PCMM Environment . . . . .	45
	Traffic Profiles . . . . .	45
	End-to-End QoS Architecture . . . . .	46
	Extending QoS to the Subscriber Edge Domain . . . . .	47
	Extending QoS to the Service Edge Domain . . . . .	47
	Provisioning End-to-End Services . . . . .	48
	Example for Videoconferencing Services . . . . .	48
	Example for Video-on-Demand Services . . . . .	49
	Using the SAE in a PCMM Environment . . . . .	50
	Logging In Subscribers and Creating Sessions . . . . .	50
	Assigned IP Subscribers . . . . .	51
	Event Notification from an IP Address Manager . . . . .	52
	SAE Communities . . . . .	53
	Storing Session Data . . . . .	54
	PCMM Record-Keeping Server Plug-In . . . . .	54
<b>Chapter 6</b>	<b>Configuring the SAE for a PCMM Environment (SRC CLI) . . . . .</b>	<b>57</b>
	Configuring the SAE for a Cable Network Environment (SRC CLI) . . . . .	57
	Configuring the SAE to Manage PCMM Devices (SRC CLI) . . . . .	58
	Setting Up SAE Communities (SRC CLI) . . . . .	61

	Configuring the SAE Community Manager . . . . .	61
	Configuring SAE Properties for the Event Notification API (SRC CLI) . . . . .	63
	Configuring Record-Keeping Server Peers for Plug-Ins (SRC CLI) . . . . .	64
	Configuring PCMM Record-Keeping Server Plug-Ins (SRC CLI) . . . . .	65
	Configuring CMTS-Specific RKS Plug-Ins (SRC CLI) . . . . .	68
<b>Chapter 7</b>	<b>Adding Objects for CMTS Devices (SRC CLI) . . . . .</b>	<b>71</b>
	Adding Objects for CMTS Devices (SRC CLI) . . . . .	71
	Creating Virtual Routers for the CMTS Device (SRC CLI) . . . . .	72
<b>Chapter 8</b>	<b>Using the NIC Resolver in a PCMM Environment . . . . .</b>	<b>75</b>
	Using the NIC Resolver in PCMM Environments . . . . .	75
<b>Part 3</b>	<b>Managing Services on RADIUS and Diameter Devices</b>	
<b>Chapter 9</b>	<b>Managing Services on Third-Party Devices in the SRC Network . . . . .</b>	<b>79</b>
	COA Script Service Overview . . . . .	79
	Configuring COA Script Services . . . . .	80
	Configuring Monitoring Agent to Receive RADIUS Accounting Messages . . . . .	80
	Creating the COA Script Service (SRC CLI) . . . . .	81
	Configuring the COA Script Service (SRC CLI) . . . . .	82
	Parameters for Sample COA Script Service . . . . .	83
	Configuring Subscriptions to the COA Script Service . . . . .	84
	Example: Using the Sample COA Script Service . . . . .	85
	Defining RADIUS Attributes for COA Requests with the API . . . . .	85
<b>Chapter 10</b>	<b>Managing the SRC Diameter Server . . . . .</b>	<b>87</b>
	Configuring the Diameter Application (SRC CLI) . . . . .	87
	Configuring the Diameter Application Properties . . . . .	87
	Configuring the Diameter Client Properties . . . . .	91
	Configuring the Diameter Server Properties . . . . .	92
	Configuring Logging Destinations . . . . .	92
	Configuring Diameter Peers (SRC CLI) . . . . .	93
	SNMP Support for Diameter Component . . . . .	96
<b>Chapter 11</b>	<b>Monitoring the SRC Diameter Server (SRC CLI) . . . . .</b>	<b>97</b>
	SRC CLI Commands to Monitor the SRC Diameter Server . . . . .	97
	Viewing Statistics for the SRC Diameter Server (SRC CLI) . . . . .	98
	Viewing Message Handler Information for the SRC Diameter Server (SRC CLI) . . . . .	98
	Viewing Server Process Information for the SRC Diameter Server (SRC CLI) . . . . .	99
	Viewing Information About SRC Diameter Server Requests (SRC CLI) . . . . .	99
	Viewing SRC Diameter Server State (SRC CLI) . . . . .	99
<b>Chapter 12</b>	<b>Managing Services with Diameter on MX Series Routers . . . . .</b>	<b>101</b>
	SRC Peer Support on MX Series Routers Overview . . . . .	101
	Managing Services on MX Series Routers Using the Diameter Application . . . . .	102
	Configuring JSRC on the MX Series Router . . . . .	103
	Configuring the Diameter Application (SRC CLI) . . . . .	103
	Configuring the Diameter Application Properties . . . . .	103
	Configuring the Diameter Client Properties . . . . .	107

	Configuring the Diameter Server Properties .....	108
	Configuring Logging Destinations .....	108
	Adding Network Devices (SRC CLI) .....	109
	Configuring Diameter Peers (SRC CLI) .....	111
	Configuring the SAE to Manage Network Devices (SRC CLI) .....	113
	Specifying Initialization Scripts for the Intelligent-Service-Edge Device Driver (SRC CLI) .....	117
	Configuring JSRC Policies (SRC CLI) .....	118
	Configuring JSRC Policy Lists .....	118
	Configuring JSRC Policy Rules .....	118
	Configuring Dynamic Profile Actions .....	119
	Configuring Operation Script for Policy Provisioning (SRC CLI) .....	120
<b>Chapter 13</b>	<b>Managing Subscriber Sessions on MX Series Routers in an SRC Network .....</b>	<b>123</b>
	Subscriber Sessions on MX Series Routers Overview .....	123
	Managing Subscriber Sessions on MX Series Routers (SRC CLI) .....	124
	Configuring External Subscriber Monitor (SRC CLI) .....	124
	Configuring Pseudo-RADIUS Authorization Server Properties (SRC CLI) ..	125
	Configuring the Pseudo-RADIUS Authorization Server (SRC CLI) ....	125
	Configuring the Directory Connection Properties for the Subscriber Data .....	128
	Configuring Directory Connection Properties for the Cached DHCP Profiles .....	129
	Configuring the NIC Proxy for the Pseudo-RADIUS Authorization Server (SRC CLI) .....	130
	Configuring Resolution Information for a NIC Proxy .....	131
	Changing the Configuration for the NIC Proxy Cache .....	131
	Configuring a NIC Proxy for NIC Replication .....	132
	Extracting RADIUS Attributes with the Pseudo-RADIUS Authorization Server (SRC CLI) .....	134
	Extracting Interface Name Attribute Values .....	134
	Extracting Virtual Router Name Attribute Values .....	135
	Enabling the Pseudo-RADIUS Authorization Server (SRC CLI) .....	137
	Disabling the Pseudo-RADIUS Authorization Server (SRC CLI) .....	137
	Setting Up MX Series Routers in the SRC Network (SRC CLI) .....	137
	Configuring the COA Script Service for MX Series Routers (SRC CLI) .....	138
	Configuring Parameters for the Script Service for MX Series Routers (SRC CLI) .....	139
	Configuring Subscriptions to the Script Service .....	141
	Viewing Statistics for the Pseudo-RADIUS Authorization Server (SRC CLI) ....	141
	Monitoring Statistics for the Pseudo-RADIUS Authorization Server (SRC CLI) .....	142



<b>Chapter 14</b>	<b>Configuring Services for SRC-Managed Routers</b> . . . . .	<b>143</b>
	DPI Script Service Overview . . . . .	143
	Creating the DPI Script Service (SRC CLI) . . . . .	144
	Configuring Subscriptions to the DPI Script Service . . . . .	145
	Parameters for DPI Script Service . . . . .	147
	Creating a Configuration File . . . . .	149
	Configuring Batch Parameters . . . . .	150
	Substituting Parameters in Policy Templates . . . . .	150
	Configuring Policy Templates . . . . .	151
	Example: Using the DPI Script Service . . . . .	156
<b>Chapter 15</b>	<b>Configuring PCC or ePCC Rules for Router Running Junos OS and Acting as PCEF</b> . . . . .	<b>159</b>
	Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF . . . . .	159
	Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF (SRC CLI) . . . . .	160
	Configuring Policies for Router Running Junos OS and Acting as PCEF (SRC CLI) . . . . .	162
	Configuring Policy Lists for Routers Running Junos OS and Acting as PCEF (SRC CLI) . . . . .	163
	Configuring Static PCC Rules for Routers Running Junos OS and Acting as PCEF (SRC CLI) . . . . .	165
	Configuring Substitutions for Gx Static PCC Rules . . . . .	168
	Configuring Dynamic PCC Rules for Routers Running Junos OS and Acting as PCEF (SRC CLI) . . . . .	168
	Configuring Substitutions for Gx Dynamic PCC Rules . . . . .	172
	Configuring the Dynamic PCC Rules Application Information for Routers Running Junos OS and Acting as PCEF (SRC CLI) . . . . .	174
	Configuring the Dynamic PCC Rules Flow Information for Routers Running Junos OS and Acting as PCEF (SRC CLI) . . . . .	175
	Configuring the Dynamic PCC Rules QoS Information for Routers Running Junos OS and Acting as PCEF (SRC CLI) . . . . .	177
	Configuring the Dynamic PCC Rules Steering Information for Routers Running Junos OS and Acting as PCEF (SRC CLI) . . . . .	178
	Configuring the Dynamic PCC Rules Redirect Information for Routers Running Junos OS and Acting as PCEF (SRC CLI) . . . . .	180
<b>Part 4</b>	<b>Using SRC Configuration Wizards</b>	
<b>Chapter 16</b>	<b>SRC Configuration Wizards Overview (SRC CLI)</b> . . . . .	<b>185</b>
	SRC Configuration Wizards Overview (SRC CLI) . . . . .	185
	How Configuration Wizards Work (SRC CLI) . . . . .	185
	Navigating Screens in the Wizard (SRC CLI) . . . . .	186
	Running a Configuration Wizard (SRC CLI) . . . . .	187

<b>Chapter 17</b>	<b>SRC Configuration Wizards Overview (C-Web Interface) . . . . .</b>	<b>189</b>
	SRC Configuration Wizards Overview (C-Web Interface) . . . . .	189
	How the Configuration Wizards Work (C-Web Interface) . . . . .	189
	Navigating Screens in the Wizard (C-Web Interface) . . . . .	190
	Running a Configuration Wizard (C-Web Interface) . . . . .	191
<b>Chapter 18</b>	<b>Using the Fair Usage on MX Series Routers Configuration Wizard . . . . .</b>	<b>193</b>
	Fair Usage on MX Series Routers Configuration Wizard Overview . . . . .	193
	Fair Usage on MX Series Routers Configuration Wizard Configuration	
	Overview . . . . .	194
	Fair Usage on MX Series Routers Configuration Wizard Definition File . . . . .	194
	Configuration Provided by the Fair Usage on MX Series Routers Configuration	
	Wizard . . . . .	201
	Required Input Parameters for the Fair Usage on MX Series Routers	
	Configuration Wizard . . . . .	204
	Running the Fair Usage on MX Series Routers Configuration Wizard (SRC	
	CLI) . . . . .	205

# List of Figures

<b>Part 1</b>	<b>Providing Specialized Services in an SRC Environment</b>	
<b>Chapter 1</b>	<b>Managing Tiered and Premium Services with QoS on JunosE Routers (SRC CLI) . . . . .</b>	<b>3</b>
	Figure 1: Searching for All QoS Profiles on a Router . . . . .	14
	Figure 2: Searching for QoS Profiles in a Policy Group . . . . .	15
	Figure 3: Searching for All Policy Groups on a Router . . . . .	15
<b>Chapter 2</b>	<b>Managing Subscribers for a Wireless Roaming Environment . . . . .</b>	<b>17</b>
	Figure 4: Subscriber Access to a Wireless Roaming Group . . . . .	18
<b>Part 2</b>	<b>Managing Services in a PCMM Environment</b>	
<b>Chapter 5</b>	<b>Providing Premium Services in a PCMM Environment . . . . .</b>	<b>41</b>
	Figure 5: PCMM Architectural Framework . . . . .	42
	Figure 6: Client Type 1 Single-Phase Resource Reservation Model . . . . .	44
	Figure 7: Client Type 2 Single-Phase Resource Reservation Model . . . . .	45
	Figure 8: SRC Software in the PCMM Environment . . . . .	45
	Figure 9: End-to-End QoS Architecture in a Cable Network . . . . .	47
	Figure 10: Videoconferencing Example . . . . .	48
	Figure 11: Video-on-Demand Example . . . . .	49
	Figure 12: Login Interactions with Assigned IP Subscribers . . . . .	51
	Figure 13: Login Interactions with Event Notification Application . . . . .	52
	Figure 14: SAE Community . . . . .	54
<b>Part 4</b>	<b>Using SRC Configuration Wizards</b>	
<b>Chapter 16</b>	<b>SRC Configuration Wizards Overview (SRC CLI) . . . . .</b>	<b>185</b>
	Figure 15: Sample SRC Configuration Wizard Screen (SRC CLI) . . . . .	186
<b>Chapter 17</b>	<b>SRC Configuration Wizards Overview (C-Web Interface) . . . . .</b>	<b>189</b>
	Figure 16: Sample SRC Configuration Wizard Screen (C-Web Interface) . . . . .	190
<b>Chapter 18</b>	<b>Using the Fair Usage on MX Series Routers Configuration Wizard . . . . .</b>	<b>193</b>
	Figure 17: Fair Usage on MX Series Routers Configuration Wizard Topology . . . . .	193
	Figure 18: SRC Host Parameters Dialog Box . . . . .	206
	Figure 19: SRC VTA Database Parameters Dialog Box . . . . .	206
	Figure 20: Router Host Parameters Dialog Box . . . . .	207



# List of Tables

	<b>About the Documentation</b> . . . . .	<b>xv</b>
	Table 1: Notice Icons . . . . .	xvi
	Table 2: Text Conventions . . . . .	xvi
<b>Part 1</b>	<b>Providing Specialized Services in an SRC Environment</b>	
<b>Chapter 1</b>	<b>Managing Tiered and Premium Services with QoS on JunosE Routers (SRC CLI)</b> . . . . .	<b>3</b>
	Table 3: Examples of Concatenated QoS Profile Input Values . . . . .	6
	Table 4: Settings for Filter Strings . . . . .	12
<b>Chapter 2</b>	<b>Managing Subscribers for a Wireless Roaming Environment</b> . . . . .	<b>17</b>
	Table 5: Packet Types for RADIUS Attributes . . . . .	21
<b>Chapter 4</b>	<b>Providing Packet Mirroring in an SRC Network</b> . . . . .	<b>29</b>
	Table 6: Parameter Substitutions for Packet-Mirroring Services . . . . .	32
<b>Part 3</b>	<b>Managing Services on RADIUS and Diameter Devices</b>	
<b>Chapter 9</b>	<b>Managing Services on Third-Party Devices in the SRC Network</b> . . . . .	<b>79</b>
	Table 7: Parameter Substitutions for COA Services . . . . .	83
<b>Chapter 11</b>	<b>Monitoring the SRC Diameter Server (SRC CLI)</b> . . . . .	<b>97</b>
	Table 8: Commands to Monitor the Diameter Server . . . . .	97
<b>Chapter 13</b>	<b>Managing Subscriber Sessions on MX Series Routers in an SRC Network</b> . . . . .	<b>123</b>
	Table 9: Parameter Substitutions for MX Series Routers COA Services . . . . .	140
<b>Chapter 14</b>	<b>Configuring Services for SRC-Managed Routers</b> . . . . .	<b>143</b>
	Table 10: Parameter Substitutions for DPI Services . . . . .	148
	Table 11: Policy Template Elements for Configuration File . . . . .	152
<b>Chapter 15</b>	<b>Configuring PCC or ePCC Rules for Router Running Junos OS and Acting as PCEF</b> . . . . .	<b>159</b>
	Table 12: Substitutions for Gx Static PCC Rules . . . . .	168
	Table 13: Substitutions for Gx Dynamic PCC Rules . . . . .	173
<b>Part 4</b>	<b>Using SRC Configuration Wizards</b>	
<b>Chapter 16</b>	<b>SRC Configuration Wizards Overview (SRC CLI)</b> . . . . .	<b>185</b>
	Table 14: Wizard Buttons (SRC CLI) . . . . .	186
	Table 15: Wizard Navigation Keys . . . . .	187

<b>Chapter 17</b>	<b>SRC Configuration Wizards Overview (C-Web Interface) . . . . .</b>	<b>189</b>
	Table 16: Wizard Pop-up Buttons . . . . .	190
<b>Chapter 18</b>	<b>Using the Fair Usage on MX Series Routers Configuration Wizard . . . . .</b>	<b>193</b>
	Table 17: SRC Configuration Parameters Supplied by the Fair Usage on MX Series Routers Configuration Wizard . . . . .	202
	Table 18: Input Parameters Required by the Fair Usage on MX Series Routers Configuration Wizard . . . . .	204

# About the Documentation

- SRC Documentation and Release Notes on page xv
- Audience on page xv
- Documentation Conventions on page xv
- Documentation Feedback on page xvii
- Requesting Technical Support on page xviii

## SRC Documentation and Release Notes

---

For a list of related SRC documentation, see <https://www.juniper.net/documentation/>.

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 OS 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 xvi](#) defines the notice icons used in this guide. [Table 2 on page xvi](#) 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.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2: Text Conventions

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



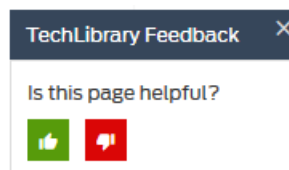
Table 2: Text Conventions (continued)

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

## Documentation Feedback

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

- Online feedback system—Click TechLibrary Feedback, on the lower right of any page on the [Juniper Networks TechLibrary](#) site, and do one of the following:



- Click the thumbs-up icon if the information on the page was helpful to you.
- Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide feedback.
- E-mail—Send your comments to [techpubs-comments@juniper.net](mailto:techpubs-comments@juniper.net). Include the document or topic name, URL or page number, and software version (if applicable).

## Requesting Technical Support

---

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

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <https://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: <https://www.juniper.net/customers/support/>
- Search for known bugs: <https://prsearch.juniper.net/>
- Find product documentation: <https://www.juniper.net/documentation/>
- Find solutions and answer questions using our Knowledge Base: <https://kb.juniper.net/>
- Download the latest versions of software and review release notes: <https://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum: <https://www.juniper.net/company/communities/>
- Create a service request online: <https://myjuniper.juniper.net>

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

## Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit <https://myjuniper.juniper.net>.
- 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 <https://support.juniper.net/support/requesting-support/>.

## PART 1

# Providing Specialized Services in an SRC Environment

- [Managing Tiered and Premium Services with QoS on JunosE Routers \(SRC CLI\) on page 3](#)
- [Managing Subscribers for a Wireless Roaming Environment on page 17](#)
- [Configuring VoIP Services in an SRC Network on page 25](#)
- [Providing Packet Mirroring in an SRC Network on page 29](#)



## CHAPTER 1

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

- [QoS on JunosE Routers Overview on page 3](#)
- [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](#)
- [Updating QoS Profile Data in the Directory on page 13](#)
- [Query Fields on page 13](#)
- [Examples: Searching for QoS Information on page 14](#)

## QoS on JunosE Routers Overview

Tiered Internet access and premium services such as video on demand, gaming, or videoconferencing require quality-of-service (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 Documentation**

- [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 13](#)

- [Examples: Searching for QoS Information on page 14](#)

## 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 website at <https://www.juniper.net/documentation/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 6](#) 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
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 Documentation**

- [QoS on JunosE Routers Overview 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 13](#)

## 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 Documentation**
- [Updating QoS Profile Data in the Directory on page 13](#)
  - [Query Fields on page 13](#)
  - [Examples: Searching for QoS Information on page 14](#)
  - [QoS on JunosE Routers Overview 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 12](#) 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: <ul style="list-style-type: none"> <li>• * (asterisk)</li> <li>• Explicit string</li> <li>• String that contains an *</li> </ul> <b>Note:</b> To define a special character (* & , !   \) in a string, precede it with the backslash symbol (\).	<ul style="list-style-type: none"> <li>• If &lt;value&gt; is *, checks for any value.</li> <li>• If &lt;value&gt; is an explicit string, checks whether any value of the property matches the string, regardless of case.</li> <li>• If &lt;value&gt; is a string that contains a *, checks whether any value of the property contains the string, regardless of case.</li> </ul>
(&<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*))
```

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 website at

<https://www.juniper.net/documentation/software/management/src>.

#### Related Documentation

- [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 13](#)
- [Examples: Searching for QoS Information on page 14](#)

## 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 Documentation

- [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 13](#)
- [QoS on JunosE Routers Overview 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 screenshot shows a dialog box titled "Router Query". It contains several input fields and a list of results.

Aspect	QoS Profile Configuration
Condition Type	Router
Condition Value	chimera
Find	QoS Profile
Supported	<input checked="" type="checkbox"/>

The following QoS Profiles are supported by Router "chimera" for QoS Profile configuration:

```

aaqp
aaqp1
atm-default
ethernet-default
serial-default
server-default
  
```

At the bottom of the dialog box are three buttons: "Query", "Clear", and "Close".



The query in [Figure 2 on page 15](#) searches for QoS profiles in policy group DHCP.

*Figure 2: Searching for QoS Profiles in a Policy Group*

The screenshot shows the 'Router Query' window with the following configuration:

- Aspect: QoS Profile Configuration
- Condition Type: Policy Group
- Condition Value: DHCP
- Find: QoS Profile
- Supported: ☒

The results pane displays the following text:

```
The following QoS Profile is supported by Policy Group "DHCP" for QoS Profile Configuration:
atm-default atm-vc atm-vp
```

At the bottom of the window are buttons for 'Query', 'Clear', and 'Close'.

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*

The screenshot shows the 'Router Query' window with the following configuration:

- Aspect: QoS Profile Configuration
- Condition Type: Router
- Condition Value: bigfoot
- Find: Policy Group
- Supported: ☒

The results pane displays the following text:

```
The following Policy Groups are supported by Router "bigfoot" for QoS Profile configuration:
content-provider (policyGroupName=content-provider,o=Policies,o=UNC)
content-provider-fast (policyGroupName=content-provider-fast,o=Policies,o=UNC)
content-provider-medium (policyGroupName=content-provider-medium,o=Policies,o=UNC)
content-provider-slow (policyGroupName=content-provider-slow,o=Policies,o=UNC)
DHCP (policyGroupName=DHCP,o=Policies,o=UNC)
eglimit (policyGroupName=eglimit,ou=ent,o=Policies,O=UNC)
EntDefault (policyGroupName=EntDefault,ou=ent,o=Policies,O=UNC)
internet-fast (policyGroupName=internet-fast,o=Policies,o=UNC)
internet-medium (policyGroupName=internet-medium,o=Policies,o=UNC)
internet-slow (policyGroupName=internet-slow,o=Policies,o=UNC)
ISP (policyGroupName=ISP,o=Policies,o=UNC)
PPP (policyGroupName=PPP,o=Policies,o=UNC)
PPP-special (policyGroupName=PPP-special,o=Policies,o=UNC)
redirect (policyGroupName=redirect,ou=ent,o=Policies,O=UNC)
```

At the bottom of the window are buttons for 'Query', 'Clear', and 'Close'.

- Related Documentation**
- [Dynamically Managing QoS Profiles on page 4](#)
  - *Policy Management Overview*
  - *Policy Components*
  - [QoS on JunosE Routers Overview on page 3](#)

## CHAPTER 2

# Managing Subscribers for a Wireless Roaming Environment

- [Wireless Roaming Environment Overview on page 17](#)
- [Subscriber Access in a Wireless Roaming Environment on page 17](#)
- [Configuring Subscriber Access for a Wireless Location on page 18](#)

## Wireless Roaming Environment Overview

---

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 a service activation engine (SAE) from any supported site.

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

### Related Documentation

- [Subscriber Access in a Wireless Roaming Environment on page 17](#)
- [Configuring Subscriber Access for a Wireless Location on page 18](#)

## Subscriber Access in a Wireless Roaming Environment

---

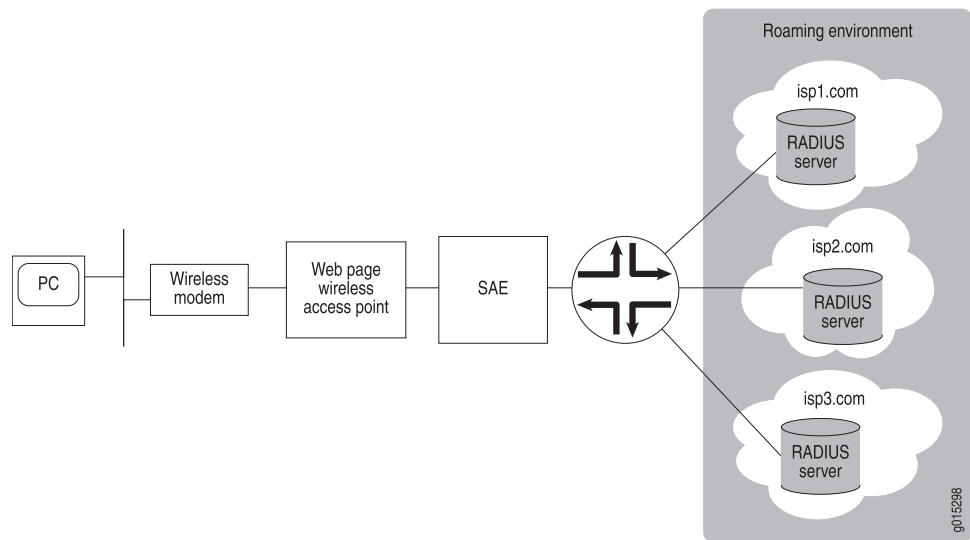
When subscribers log in to a wireless location that has a roaming agreement with other locations, the following sequence of events occurs:

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 Documentation**
- [Wireless Roaming Environment Overview 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 website at:  
<https://www.juniper.net/documentation/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-fi-alliance.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`

- For each attribute that you configure, configure the packet type to which the attribute applies. [Table 5 on page 21](#) 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
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 webpage to which subscribers connect. The policies associated with the service should specify a Junos OS 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:

- Create the SRC service to use RADIUS authentication.  
*See Adding a Normal Service (SRC CLI).*
- Create a policy group that 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)*

- *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 webpage 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 website at

<https://www.juniper.net/documentation/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 message 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 Documentation

- [Wireless Roaming Environment Overview on page 17](#)
- [Subscriber Access in a Wireless Roaming Environment on page 17](#)



## CHAPTER 3

# Configuring VoIP Services in an SRC Network

- [Session Management for VoIP Services Overview on page 25](#)
- [Configuring Policies and Services for VoIP on page 26](#)
- [Activating VoIP Services for Assigned IP Subscribers on page 27](#)
- [Setting Timeouts for Assigned IP Subscriber Sessions on page 28](#)

### Session Management for VoIP Services Overview

---

When the service activation engine (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.

For information about configuring and managing policies, see the *SRC PE Services and Policies Guide*.

### Related Documentation

- [Global and Local Parameters Overview](#)
- [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 Documentation**
- [Session Management for VoIP Services Overview on page 25](#)
  - [Configuring Policy Groups \(SRC CLI\)](#)
  - [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 Documentation**
- [Session Management for VoIP Services Overview 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 Documentation

- [Session Management for VoIP Services Overview 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

- [Packet-Mirroring Services Overview on page 29](#)
- [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](#)
- [Specifying Maximum Number of RADIUS Peers \(SRC CLI\) on page 34](#)
- [Example: Using the Sample Packet-Mirroring Application on page 35](#)
- [Defining RADIUS Attributes for Dynamic Authorization Requests with the SAE Core API on page 37](#)

### Packet-Mirroring Services Overview

---

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 service activation engine (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 Documentation**
- [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](#).

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).

- Related Documentation**
- [Configuring the Script Service for Packet Mirroring on page 31](#)
  - [Example: Using the Sample Packet-Mirroring Application on page 35](#)
  - [Packet-Mirroring Services Overview 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.smgmt.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.smgmt.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 Documentation

- [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
- [SRC Script Services Overview](#)
- [Packet-Mirroring Services Overview](#) 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.

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

Parameter Name	Description
dynRetry	Number of retries for sending dynamic RADIUS packet when no RADIUS response is received. The retry interval is 3 seconds.
dynConfig	<p>Content of dynamic RADIUS request packets in the format &lt;action&gt;. &lt;radiusAttributeName&gt;=&lt;pluginEventAttribute&gt;\n</p> <ul style="list-style-type: none"> <li>action—Action that is executed on packet content (attribute) <ul style="list-style-type: none"> <li>start</li> <li>stop</li> <li>start-stop</li> </ul> </li> <li>radiusAttributeName—Valid RADIUS attribute specified as follows: <ul style="list-style-type: none"> <li>Standard RADIUS attribute name or number.</li> <li>JunosE VSA in one of the following formats: vendor-specific.4874.&lt;vsa#&gt;[.salt] 26.4874.&lt;vsa#&gt;[.salt] where .salt indicates that the attribute is MD5 salt-encrypted in the RADIUS packet.</li> </ul> </li> <li>pluginEventAttribute—Valid Python expression</li> <li>\n—New-line character included between the lines of a configuration containing multiple lines; the entire configuration must be enclosed in quotation marks</li> </ul> <p>For example:</p> <pre>start-stop.Acct-Session-Id = ifSessionId "start-stop.Acct-Session-Id=ifSessionId\n start.vendor-specific.JUNIPER.Unisphere-Med-Dev-Handle.salt= custom['dynMirrorIdentifier']\n start.vendor-specific. JUNIPER.Unisphere-Med-Ip-Address.salt= intIp(custom['dynAnalyzerIpAddress'])\n start.vendor-specific. JUNIPER.Unisphere-Med-Port-Number.salt= int(custom ['dynAnalyzerPortNumber'])\n stop.vendor- specific.4874.58.salt=0"</pre>

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.sgmt.scriptServices.packetMirroring.LiService;
  file file:///opt/UMC/sae/lib/pm.jar;
}
```

#### Related Documentation

- [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 37](#)

---

## 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 Documentation

- [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 37](#)
- [Example: Using the Sample Packet-Mirroring Application on page 35](#)
- [Packet-Mirroring Services Overview 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 website:

<https://www.juniper.net/support/downloads/?p=src#sw>

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 website:

<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 37](#).

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

For information about subscriptions, see *Subscriptions Overview*.

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 class path 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:

```
[retailername=default,o=Users,o=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 Documentation**
- [Configuring Packet-Mirroring Support in an SRC Network on page 30](#)
  - [Packet-Mirroring Services Overview 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.

For information about the `ServiceSessionInfo` interface, see the script service documentation in the SAE core API documentation on the Juniper Networks website at

<https://www.juniper.net/documentation/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/smgmt/scriptServices/packetMirroring/LiService.java`.

- Related Documentation**
- [Configuring Parameters for the Script Service for Packet Mirroring on page 32](#)





## PART 2

# Managing Services in a PCMM Environment

- [Providing Premium Services in a PCMM Environment on page 41](#)
- [Configuring the SAE for a PCMM Environment \(SRC CLI\) on page 57](#)
- [Adding Objects for CMTS Devices \(SRC CLI\) on page 71](#)
- [Using the NIC Resolver in a PCMM Environment on page 75](#)



## CHAPTER 5

# Providing Premium Services in a PCMM Environment

- [PCMM Environment Overview on page 41](#)
- [Using the SAE in a PCMM Environment on page 50](#)

### PCMM Environment Overview

---

The PacketCable Multimedia (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

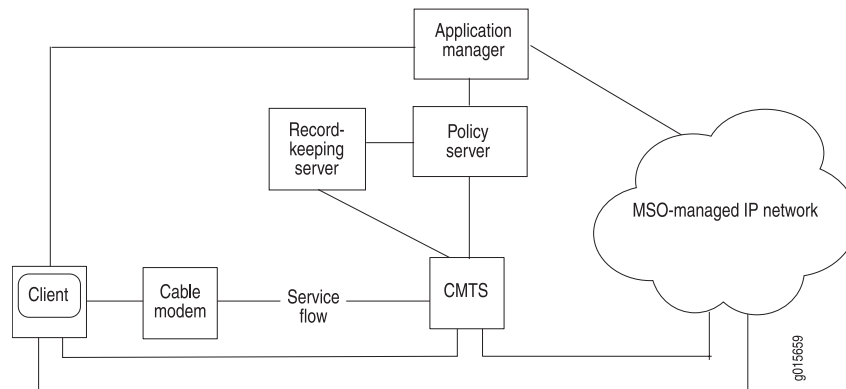
### PCMM Architecture

[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 head-end 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 head-end to cable modems.
- Upstream channels—Carry signals from the cable modems to the CMTS head-end.

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.

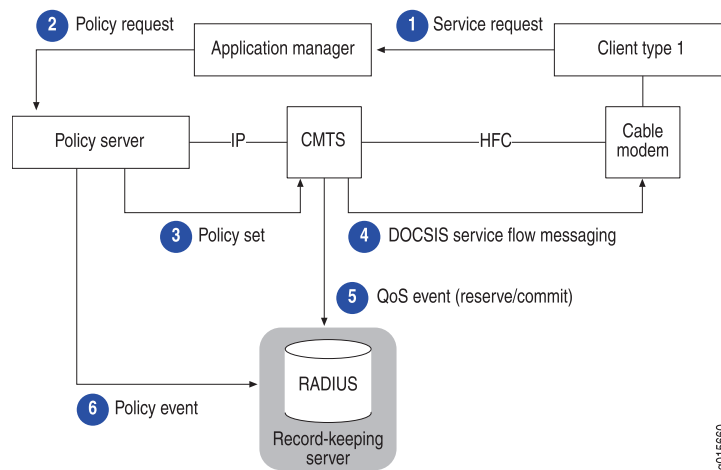
The PCMM specification defines two resource reservation models for each client type—a single phase and a dual phase. The SRC software supports the single-phase model.

### ***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.

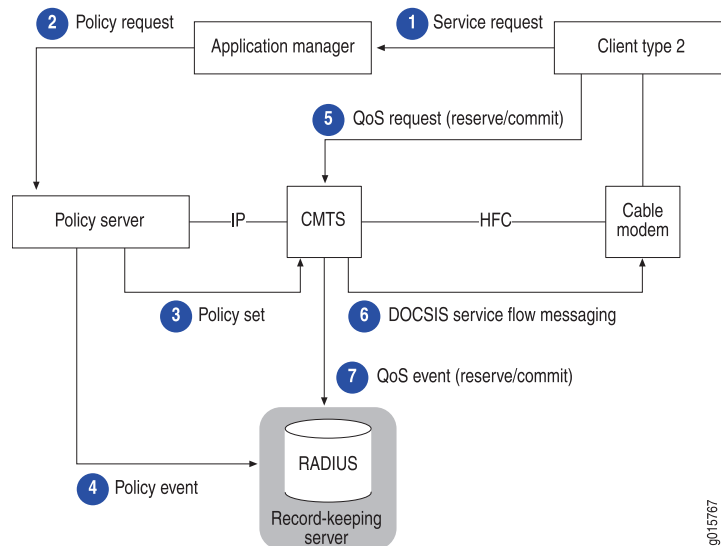
Figure 6 on page 44 shows the message flow in an application scenario for the client type 1 single-phase resource reservation model.

**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.

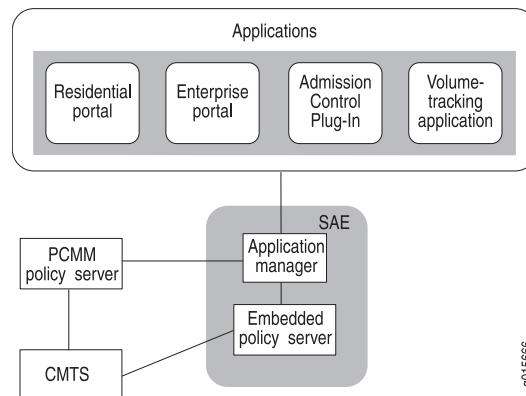
Client type 2 entities support the client-requested QoS with policy-push scenario of service delivery defined in PacketCable Multimedia Architecture Framework Technical Report (PKT-TR-MM-ARCH). In this scenario, the application manager requests QoS resources on behalf of the client, and the policy server 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.

**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

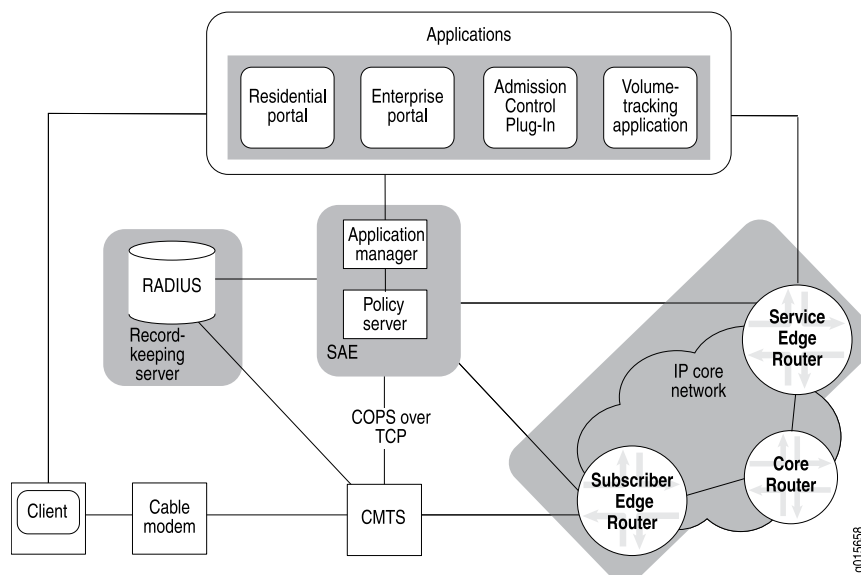
The previous sections show how the SRC software supports QoS in the cable operator's RF domain, which encompasses the connection from the cable modem to the CMTS device. Using the SRC software along with Juniper Networks routers, you can link the RF domain to the subscriber and service edge domains.

- 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 OS–based firewall services
- Routers running Junos OS 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 OS–based firewall services
- Routers running Junos OS VPN services

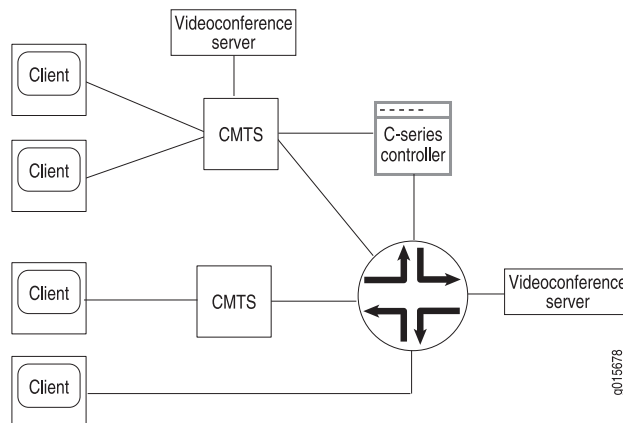
### Provisioning End-to-End Services

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:

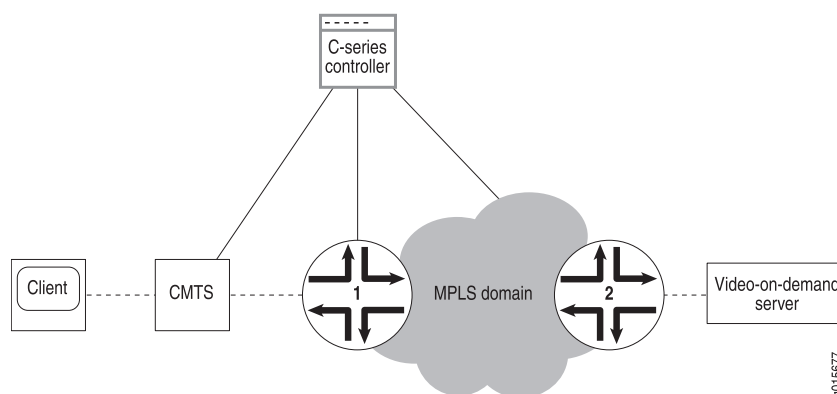
- Three services:
  - A service that provides policies to mark packets with a specified type of service for the videoconferencing software.
  - A service that provides policies for the type of service specified for CMTS device.
  - A service that provides policies for the type of service specified for 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 OS 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 OS
  - 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 OS, the MPLS domain, and the CMTS device, and sets up the MPLS path from router running Junos OS (2) to router running Junos OS (1).

#### Related Documentation

- 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
- PacketCable Security Specifications (PKT-SP-SEC)
- [Using the SAE in a PCMM Environment on page 50](#)
- [Using the NIC Resolver in PCMM Environments on page 75](#)
- *Example: Providing Premium Services*

---

## 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 OS.

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

- [Logging In Subscribers and Creating Sessions on page 50](#)
- [SAE Communities on page 53](#)
- [Storing Session Data on page 54](#)

## 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.

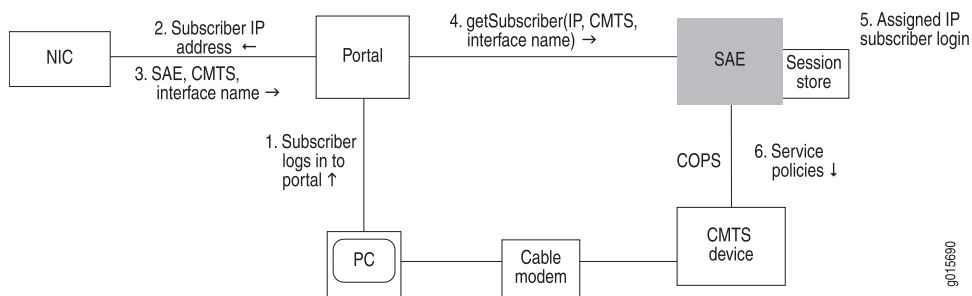
### Assigned IP Subscribers

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

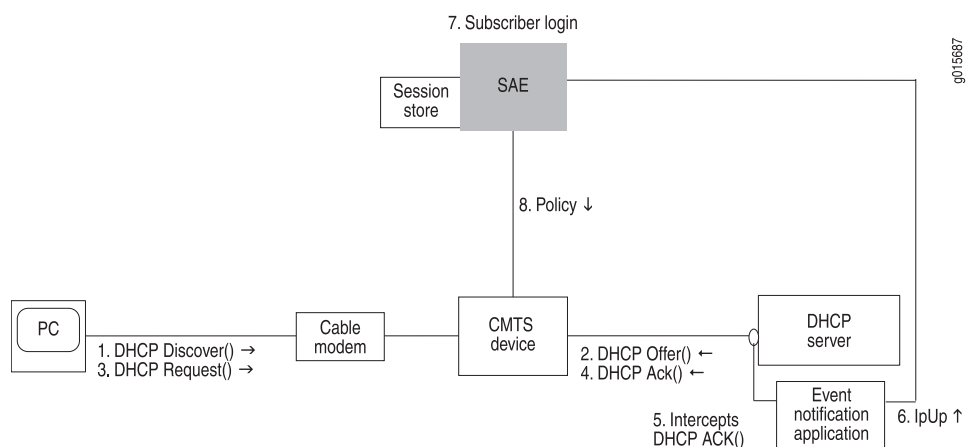
With the event notification method of logging in subscribers and creating subscriber sessions, the subscriber logs in to the CMTS device and obtains an IP address through an address server, usually a DHCP server. The SAE receives notifications about the subscriber, such as the subscriber's IP address, 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 event notification application or the Monitoring Agent that monitors DHCP traffic sends an ipUp event for each Ack message sent from the DHCP server 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 event notification application or the Monitoring Agent that monitors DHCP traffic 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.



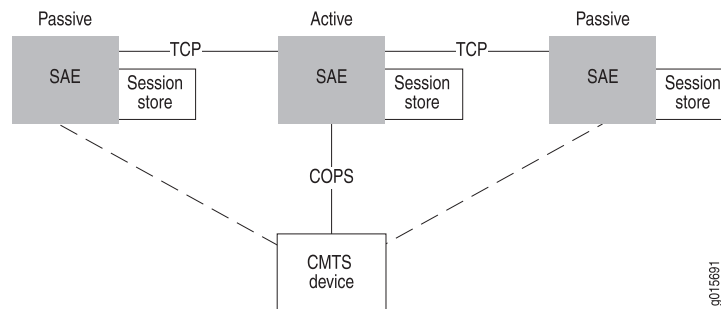
**NOTE:** To prefer the second user session for an existing address upon receiving an ipUp event, set the **prefer-second-user-session** option under the **[edit shared sae configuration driver third-party]** hierarchy.

## 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 Documentation

- [PCMM Environment Overview on page 41](#)
- [Using the NIC Resolver in PCMM Environments on page 75](#)



- [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 \(SRC CLI\)](#) on page 57
- [Configuring the SAE to Manage PCMM Devices \(SRC CLI\)](#) on page 58
- [Setting Up SAE Communities \(SRC CLI\)](#) on page 61
- [Configuring the SAE Community Manager](#) on page 61
- [Configuring SAE Properties for the Event Notification API \(SRC CLI\)](#) on page 63
- [Configuring Record-Keeping Server Peers for Plug-Ins \(SRC CLI\)](#) on page 64
- [Configuring PCMM Record-Keeping Server Plug-Ins \(SRC CLI\)](#) on page 65
- [Configuring CMTS-Specific RKS Plug-Ins \(SRC CLI\)](#) on page 68

## 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 63 (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 64 (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 65 (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 68.

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 71.

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

See *Using the NIC Resolver*.

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 Documentation

- [PCMM Environment Overview 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-io3-policy disable-pcmm-io3-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 PcmCommunityManager;
disable-full-sync true;
disable-pcmm-i03-policy true;
session-recovery-retry-interval 3600000;
element-id 1;
default-rks-plugin-in rksTracking;
```

- Related Documentation**
- [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 68](#)
  - *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 72](#).
- 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 Documentation**
- [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 63](#)

## 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 Documentation**
- [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
EventAPI]
```

```
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 Documentation

- [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 Documentation

- [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 65](#)
- [Configuring CMTS-Specific RKS Plug-Ins \(SRC CLI\) on page 68](#)
- [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);
  fallback-timer fallback-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 64](#).

```
[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 68](#).

- Related Documentation**
- [Using the SAE in a PCMM Environment on page 50](#)
  - [PCMM Environment Overview 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 Documentation**
- [Configuring CMTS-Specific RKS Plug-Ins \(C-Web Interface\)](#)
  - [Configuring Record-Keeping Server Peers for Plug-Ins \(SRC CLI\) on page 64](#)
  - [Configuring PCMM Record-Keeping Server Plug-Ins \(SRC CLI\) on page 65](#)
  - [Adding Objects for CMTS Devices \(SRC CLI\) on page 71](#)

- *Initially Configuring the SAE*





## CHAPTER 7

# Adding Objects for CMTS Devices (SRC CLI)

- [Adding Objects for CMTS Devices \(SRC CLI\) on page 71](#)
- [Creating Virtual Routers for the CMTS Device \(SRC CLI\) on page 72](#)

## 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. 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  
Documentation**

- [Connections to Managed Devices](#)
- [Configuring CMTS-Specific RKS Plug-Ins \(SRC CLI\) on page 68](#)
- [Creating Virtual Routers for the CMTS Device \(SRC CLI\) on page 72](#)

---

## 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. 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-plugin [ tracking-plugin ...]
```

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-plugin rksPlugin;
```

**Related  
Documentation**

- [Adding Objects for CMTS Devices \(SRC CLI\) on page 71](#)
- [Configuring CMTS-Specific RKS Plug-Ins \(SRC CLI\) on page 68](#)
- [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 on page 75](#)

## 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 VrSaeld collects and publishes information about the mappings of policy server groups to SAEs.

### Related Documentation

- [PCMM Environment Overview 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*



## PART 3

# Managing Services on RADIUS and Diameter Devices

- [Managing Services on Third-Party Devices in the SRC Network on page 79](#)
- [Managing the SRC Diameter Server on page 87](#)
- [Monitoring the SRC Diameter Server \(SRC CLI\) on page 97](#)
- [Managing Services with Diameter on MX Series Routers on page 101](#)
- [Managing Subscriber Sessions on MX Series Routers in an SRC Network on page 123](#)
- [Configuring Services for SRC-Managed Routers on page 143](#)
- [Configuring PCC or ePCC Rules for Router Running Junos OS and Acting as PCEF on page 159](#)





## CHAPTER 9

# Managing Services on Third-Party Devices in the SRC Network

- [COA Script Service Overview on page 79](#)
- [Configuring COA Script Services on page 80](#)
- [Configuring Monitoring Agent to Receive RADIUS Accounting Messages on page 80](#)
- [Creating the COA Script Service \(SRC CLI\) on page 81](#)
- [Configuring the COA Script Service \(SRC CLI\) on page 82](#)
- [Parameters for Sample COA Script Service on page 83](#)
- [Configuring Subscriptions to the COA Script Service on page 84](#)
- [Example: Using the Sample COA Script Service on page 85](#)
- [Defining RADIUS Attributes for COA Requests with the API on page 85](#)

## COA Script Service Overview

---

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

### Related Documentation

- [Configuring COA Script Services on page 80](#)
- [Configuring Subscriptions to the COA Script Service on page 84](#)
- [Configuring Monitoring Agent to Receive RADIUS Accounting Messages on page 80](#)
- [Parameters for Sample COA Script Service on page 83](#)
- [Example: Using the Sample COA Script Service on page 85](#)

## 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 80](#)
- [“Creating the COA Script Service \(SRC CLI\)” on page 81](#)
- [“Configuring the COA Script Service \(SRC CLI\)” on page 82](#)
- [“Configuring Subscriptions to the COA Script Service” on page 84](#)

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 85](#).

### Related Documentation

- [COA Script Service Overview on page 79](#)
- [Defining RADIUS Attributes for COA Requests with the API on page 85](#)
- [Setting Up Script Services](#)
- [Parameters for Sample COA Script Service on page 83](#)

## Configuring Monitoring Agent to Receive RADIUS Accounting Messages

---

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

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

- Make sure there is no other RADIUS server listening on the RADIUS accounting port, and enable the `MonAgent.radius.server` property.

- Configure the shared secret between the RADIUS server and the RADIUS client by specifying the `MonAgent.radius.secret.<IP address>` property.

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

**Related  
Documentation**

- [Configuring the COA Script Service \(SRC CLI\) on page 82](#)
- [Defining RADIUS Attributes for COA Requests with the API on page 85](#)

## 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.smgmt.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.smgmt.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.smgmt.sae.coa.CoaService;
  file file:///opt/UMC/sae/var/run/coa.jar;
}
```

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

#### Related Documentation

- [COA Script Service Overview on page 79](#)
- [Configuring Subscriptions to the COA Script Service on page 84](#)
- [Configuring COA Script Services on page 80](#)
- [Configuring the COA Script Service \(SRC CLI\) on page 82](#)
- [Parameters for Sample COA Script Service on page 83](#)

---

## 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 83](#).

#### Related Documentation

- [COA Script Service Overview on page 79](#)
- [Configuring Subscriptions to the COA Script Service on page 84](#)
- [Creating the COA Script Service \(SRC CLI\) on page 81](#)
- [Configuring COA Script Services on page 80](#)
- [Example: Using the Sample COA Script Service on page 85](#)

## Parameters for Sample COA Script Service

[Table 7 on page 83](#) 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
<code>dynClientIp</code>	IP address of the third-party device.
<code>dynClientPort</code>	UDP port number of the third-party device.
<code>dynServerIp</code>	IP address of the C Series Controller.
<code>dynServerPort</code>	UDP port number of the C Series Controller.
<code>dynSecret</code>	Shared secret between RADIUS server and RADIUS client.
<code>dynRetry</code>	Number of retries for sending COA messages when no RADIUS response is received. The retry interval is 3 seconds.

*Table 7: Parameter Substitutions for COA Services (continued)*

Parameter Name	Description
dynConfig	<p>Content of service definition in the format  &lt;action&gt;. &lt;radiusAttributeName&gt;=&lt;pluginEventAttribute&gt;\n</p> <ul style="list-style-type: none"> <li>• action—Action that is executed on packet content (attribute): <ul style="list-style-type: none"> <li>• start</li> <li>• stop</li> <li>• start-stop</li> </ul> </li> <li>• radiusAttributeName—Valid RADIUS attribute specified as follows: <ul style="list-style-type: none"> <li>• Standard RADIUS attribute name or number</li> <li>• Third-party VSA in the format  vendor-specific.&lt;vendor#&gt;.&lt;vsa#&gt;.string</li> </ul> </li> <li>• pluginEventAttribute—Valid expression in the format: <ul style="list-style-type: none"> <li>• Python expression</li> <li>• &lt;commandCode&gt;&lt;serviceName&gt;; the entire expression must be enclosed in single quotation marks and you must use three backslashes (\\\) to escape the backslash that starts a &lt;commandCode&gt;  For example: \x0b would be replaced by \\x0b</li> </ul> </li> <li>• \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"</li> </ul>

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 85](#)).

#### Related Documentation

- [COA Script Service Overview on page 79](#)
- [Configuring Monitoring Agent to Receive RADIUS Accounting Messages on page 80](#)
- [Creating the COA Script Service \(SRC CLI\) on page 81](#)
- [Configuring COA Script Services on page 80](#)
- [Example: Using the Sample COA Script Service on page 85](#)

## Configuring Subscriptions to the COA Script Service

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

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

- Related Documentation**
- [COA Script Service Overview on page 79](#)
  - [Configuring COA Script Services on page 80](#)
  - [Configuring the COA Script Service \(SRC CLI\) on page 82](#)
  - [Example: Using the Sample COA Script Service on page 85](#)

---

## 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/BODIM.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 BODIM 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 82](#). For information about passing the values through the SAE core API, see [“Defining RADIUS Attributes for COA Requests with the API” on page 85](#).

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

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

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

- Related Documentation**
- [COA Script Service Overview on page 79](#)
  - [Configuring Subscriptions to the COA Script Service on page 84](#)
  - [Configuring COA Script Services on page 80](#)
  - [Creating the COA Script Service \(SRC CLI\) on page 81](#)

---

## 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 82](#))

- 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 website at <https://www.juniper.net/documentation/software/management/src/api-index.html>.

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

**SDK/scriptServices/coa/java/net/juniper/smgmt/scriptServices/coa/CoaService.java.**

**Related  
Documentation**

- [COA Script Service Overview on page 79](#)
- [Configuring COA Script Services on page 80](#)
- [Creating the COA Script Service \(SRC CLI\) on page 81](#)
- [Configuring Monitoring Agent to Receive RADIUS Accounting Messages on page 80](#)



## CHAPTER 10

# Managing the SRC Diameter Server

- [Configuring the Diameter Application \(SRC CLI\) on page 87](#)
- [Configuring Diameter Peers \(SRC CLI\) on page 93](#)
- [SNMP Support for Diameter Component on page 96](#)

### Configuring the Diameter Application (SRC CLI)

---

You can configure the properties of the application, client, server, and logging destination of the SRC Diameter application.

Perform the following tasks to configure these properties:

- [Configuring the Diameter Application Properties on page 87](#)
- [Configuring the Diameter Client Properties on page 91](#)
- [Configuring the Diameter Server Properties on page 92](#)
- [Configuring Logging Destinations on page 92](#)

### Configuring the Diameter Application Properties

The SRC software supports Diameter application properties such as Juniper Networks Session Resource Control (JSRC) and southbound Gx interface. JSRC and southbound Gx interface communicate with the Service Activation Engine (SAE) (remote SRC peer).

Use the following configuration statements to configure the properties for the Diameter application:

```
system diameter {  
  java-heap-size java-heap-size;  
  java-new-size java-new-size;  
  java-garbage-collection-options java-garbage-collection-options;  
  protocol [(tcp | sctp)...];  
  local-address [local-address...];  
  port port;  
  origin-host origin-host;  
  origin-realm origin-realm;  
  diameter-server-timeout diameter-server-timeout;  
  active-peers;  
  debug-mode;  
  load-balancing-mode (failover | round-robin);  
}
```

```
transaction-processing-log (log-no-messages | log-severe-messages |
    log-normal-messages | log-debug-messages);
packet-trace-log (log-no-messages | log-severe-messages | log-normal-messages |
    log-debug-messages);
peer-state-machine-log (log-no-messages | log-severe-messages | log-normal-messages
    | log-debug-messages);
configuration-log (log-no-messages | log-severe-messages | log-normal-messages |
    log-debug-messages);
}
```

To configure the Diameter application:

1. From configuration mode, access the statement for the Diameter application.

```
user@host# edit system diameter
```



**NOTE:** The java-\* options have default values that should not be changed unless directed by Juniper Networks Technical Assistance Center (JTAC).

2. If you encounter problems caused by lack of memory, change the maximum memory size available to the Java Runtime Environment (JRE).

```
[edit system diameter]
user@host# set java-heap-size java-heap-size
```

3. Configure the amount of space available to the JRE when the Diameter server starts.

```
[edit system diameter]
user@host# set java-new-size java-new-size
```

4. Configure the garbage collection functionality of the Java Virtual Machine.

```
[edit system diameter]
user@host# set java-garbage-collection-options java-garbage-collection-options
```

5. Specify the protocol for the transport connection.

```
[edit system diameter]
user@host# set protocol [(tcp | sctp)...
```

6. (Optional) Specify the local IP addresses that remote peers can use to reach this server.

```
[edit system diameter]
user@host# set local-address [local-address...]
```

7. (Optional) Specify the port for the server.

```
[edit system diameter]
user@host# set port port
```

8. (Optional) Specify the fully qualified domain name (FQDN) used to identify this host to its Diameter peers.

```
[edit system diameter]
user@host# set origin-host origin-host
```

9. (Optional) Specify the realm used to identify this host to its Diameter peers.

```
[edit system diameter]
user@host# set origin-realm origin-realm
```

The Diameter realm should be configured to the domain name of the origin host. For example, if the FQDN of the host is host.juniper.net, then the realm should be juniper.net.

10. (Optional) Configure the timeout value until which the Diameter server holds unsolicited requests such as Point to Point Protocol (PPP) and Abort Session Request (ASR), and waits for a matching response such as Push Profile Answer (PPA) and Abort Session Answer (ASA). The server discards the responses received after the specified time. The value range is 1–65,565 seconds. The preferred value is 10–30 seconds. By default, the value is set to 25 seconds.

```
[edit system diameter]
user@host# set diameter-server-timeout diameter-server-timeout
```



**NOTE:** `diameter-server-timeout` and `reply-timeout` under the `[edit shared sae group configuration driver]` hierarchy should be configured with the same value.

11. (Optional) Specify whether the peer connection is in active mode.

```
[edit system diameter]
user@host# set active-peers
```



---

**NOTE:**

- Active mode means that the SRC software actively tries to connect to the peer. Make sure the peer you are connecting to supports active peers. The MX Series router does not support active peers. The SRC software can still be configured, but the connection attempts will not work.
  - If the peer connection is configured to be in active mode, you must configure the remote peer address for all Diameter peers by using the **address** option under the **[edit shared network diameter peer *name*]** hierarchy.
- 

12. (Optional) Specify whether the peer connection is in debug mode.

```
[edit system diameter]
user@host# set debug-mode
```

13. (Optional) Configure the load-balancing mode for peer selection when forwarding a request message.

```
[edit system diameter]
user@host# set load-balancing-mode (failover | round-robin)
```

14. (Optional) Configure the log level for the transaction processing log.

```
[edit system diameter]
user@host# set transaction-processing-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

15. (Optional) Configure the log level for the packet tracing log.

```
[edit system diameter]
user@host# set packet-trace-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

16. (Optional) Configure the log level for the peer state machine log.

```
[edit system diameter]
user@host# set peer-state-machine-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

17. (Optional) Configure the log level for the configuration log.

```
[edit system diameter]
user@host# set configuration-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

## Configuring the Diameter Client Properties

This procedure configures the client-side adapter of the SRC Diameter server, which handles client connections. Configuration should be necessary only if you encounter performance problems.

Use the following statements to configure the properties for the Diameter client:

```
system diameter client {
  threads threads;
  keep-alive-time keep-alive-time;
}
```

To configure the Diameter client properties:

1. From configuration mode, access the statement for the Diameter client.

```
user@host# edit system diameter client
```

2. (Optional) Specify the number of threads to use.

```
[edit system diameter client]
user@host# set threads threads
```

3. (Optional) Specify the time to wait for new commands.

```
[edit system diameter client]
user@host# set keep-alive-time keep-alive-time
```

- See Also**
- [Configuring the Diameter Server Properties on page 92](#)
  - [Configuring Logging Destinations on page 92](#)

## 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
```

- See Also**
- [Configuring the Diameter Client Properties on page 91](#)
  - [Configuring Logging Destinations on page 92](#)

## Configuring Logging Destinations

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

```
system diameter logger name ...
```

```

system diameter logger name file {
  filter filter;
  filename filename;
  rollover-filename rollover-filename;
  maximum-file-size maximum-file-size;
}

```

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

1. From configuration mode, access the statement that configures the name and type of logging destination.

```

user@host# edit system diameter logger name file

```

2. Specify the properties for the logging destination.

```

[edit system diameter logger name file]
user@host# set ?

```

For more information about configuring properties for the logging destination, see *Configuring Logging Destinations to Store Messages in a File (SRC CLI)*.

- Related Documentation**
- [SRC CLI Commands to Monitor the SRC Diameter Server on page 97](#)
  - To manage services for JSRC peers on MX Series routers, see [Managing Services on MX Series Routers Using the Diameter Application on page 102](#).

## Configuring Diameter Peers (SRC CLI)

Use the following configuration statements to configure the Diameter peers:

```

shared network diameter peer name {
  protocol [(tcp | sctp)...];
  address [address...];
  enforce-source-address;
  local-address local-address;
  connect-timeout connect-timeout;
  watchdog-timeout watchdog-timeout;
  state-machine-timeout state-machine-timeout;
  reconnect-timeout reconnect-timeout;
  port port;
  origin-host origin-host;
  incoming-queue-limit incoming-queue-limit;
  active-peer;
}

```



**NOTE:** When you commit the Diameter peer configuration, keep in mind the following conditions:

- The origin host, remote peer address, or both should be specified for the Diameter peer.
- If the enforce source address is configured for the Diameter peer, the remote peer address should be specified for the Diameter peer.
- If the peer connection is configured to be in active mode for a particular Diameter peer or globally for all Diameter peers by using the **active-peers** option under the **[edit system diameter]** hierarchy, the remote peer address should be specified for the Diameter peers.

---

To configure the Diameter peer:

1. From configuration mode, access the statements for the peer.

```
user@host# edit shared network diameter peer name
```

The peer name must be unique.

2. Specify the protocol for the transport connection.

```
[edit shared network diameter peer name]  
user@host# set protocol [(tcp | sctp)...] 
```

3. (Optional) Specify the addresses of the remote peer. If SCTP is the transport protocol, you can specify multiple addresses. If TCP is the transport protocol, you can specify only a single address.

```
[edit shared network diameter peer name]  
user@host# set address [address...] 
```

4. (Optional) Specify whether the remote peer must connect from one of the IP addresses listed by the **address** option.

```
[edit shared network diameter peer name]  
user@host# set enforce-source-address 
```

5. (Optional) Specify the local address of the peer.

```
[edit shared network diameter peer name]  
user@host# set local-address local-address 
```

6. (Optional) Specify the maximum amount of time allowed for the Diameter peer to respond to a connection request.



```
[edit shared network diameter peer name]  
user@host# set connect-timeout connect-timeout
```

7. (Optional) Specify the watchdog timeout used for the connection to the remote peer.

```
[edit shared network diameter peer name]  
user@host# set watchdog-timeout watchdog-timeout
```

8. (Optional) Specify the Diameter state machine timeout.

```
[edit shared network diameter peer name]  
user@host# set state-machine-timeout state-machine-timeout
```

9. (Optional) Specify the time interval between connection attempts when the peer is in the disconnected state.

```
[edit shared network diameter peer name]  
user@host# set reconnect-timeout reconnect-timeout
```

10. (Optional) Specify the port for the client.

```
[edit shared network diameter peer name]  
user@host# set port port
```

11. (Optional) Specify the identifier for the endpoint that the peer presents during connection establishment.

```
[edit shared network diameter peer name]  
user@host# set origin-host origin-host
```

12. (Optional) Specify the number of messages allowed on the incoming message queue for a peer.

```
[edit shared network diameter peer name]  
user@host# set incoming-queue-limit incoming-queue-limit
```

13. (Optional) Specify whether the peer connection is in active mode.

```
[edit shared network diameter peer name]  
user@host# set active-peer
```



**NOTE:** Active mode means that the SRC software actively tries to connect to the peer. Make sure the peer you are connecting to supports active peers. The MX Series router does not support active peers. The SRC software can still be configured, but the connection attempts will not work.

---

**Related  
Documentation**

- [Configuring the Diameter Application \(SRC CLI\) on page 87](#)
- [Viewing SRC Diameter Server State \(SRC CLI\) on page 99](#)

---

## SNMP Support for Diameter Component

---

You can monitor the statistics and status of Diameter components by using the Diameter MIB. The SNMP support is available for the Diameter component information which can be retrieved with SNMP commands. You can query the MIB through any of the SNMP commands.

The name of the MIB for Diameter component is Juniper-SDX-DIAMETER-MIB. You can access the MIBs on the Juniper website at

<https://www.juniper.net/documentation/software/management/src>

**Related  
Documentation**

- [Configuring the Diameter Application \(SRC CLI\) on page 87](#)
- [Viewing SRC Diameter Server State \(SRC CLI\) on page 99](#)
- [Viewing Statistics for the SRC Diameter Server \(SRC CLI\) on page 98](#)

## CHAPTER 11

# Monitoring the SRC Diameter Server (SRC CLI)

- [SRC CLI Commands to Monitor the SRC Diameter Server on page 97](#)
- [Viewing Statistics for the SRC Diameter Server \(SRC CLI\) on page 98](#)
- [Viewing Message Handler Information for the SRC Diameter Server \(SRC CLI\) on page 98](#)
- [Viewing Server Process Information for the SRC Diameter Server \(SRC CLI\) on page 99](#)
- [Viewing Information About SRC Diameter Server Requests \(SRC CLI\) on page 99](#)
- [Viewing SRC Diameter Server State \(SRC CLI\) on page 99](#)

## SRC CLI Commands to Monitor the SRC Diameter Server

---

You can view statistics and status for the SRC Diameter server. [Table 8 on page 97](#) lists the commands you use to monitor the SRC Diameter server

*Table 8: Commands to Monitor the Diameter Server*

Command	Output Displayed
<code>show diameter statistics</code>	Information about the server process and the current state of the Diameter server.
<code>show diameter statistics message-handler</code>	Information about the Diameter server message handler.
<code>show diameter statistics message-handler message-flow</code>	Information about the Diameter server message flows.
<code>show diameter statistics process</code>	Information about the Diameter server process.
<code>show diameter statistics requests</code>	Information about the Diameter server requests.
<code>show diameter status</code>	Status of the Diameter server.
<code>show diameter status clients</code>	Status of the Diameter clients.
<code>show diameter status peers</code>	Status of the Diameter peers.

- Related Documentation**
- [Configuring the Diameter Application \(SRC CLI\) on page 87](#)
  - [Viewing Statistics for the SRC Diameter Server \(SRC CLI\) on page 98](#)
  - [Viewing Message Handler Information for the SRC Diameter Server \(SRC CLI\) on page 98](#)
  - [Viewing Server Process Information for the SRC Diameter Server \(SRC CLI\) on page 99](#)
  - [Viewing Information About SRC Diameter Server Requests \(SRC CLI\) on page 99](#)
  - [Viewing SRC Diameter Server State \(SRC CLI\) on page 99](#)

---

## Viewing Statistics for the SRC 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
```

- Related Documentation**
- [Configuring the Diameter Application \(SRC CLI\) on page 87](#)
  - [SRC CLI Commands to Monitor the SRC Diameter Server on page 97](#)

---

## Viewing Message Handler Information for the SRC 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
```

- Related Documentation**
- [Configuring the Diameter Application \(SRC CLI\) on page 87](#)
  - [SRC CLI Commands to Monitor the SRC Diameter Server on page 97](#)

## Viewing Server Process Information for the SRC Diameter Server (SRC CLI)

---

**Purpose** View information about the server process.

**Action** To display about the server process:

```
user@host> show diameter statistics process
```

**Related Documentation**

- [Configuring the Diameter Application \(SRC CLI\) on page 87](#)
- [SRC CLI Commands to Monitor the SRC Diameter Server on page 97](#)

## Viewing Information About SRC 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
```

**Related Documentation**

- [Configuring the Diameter Application \(SRC CLI\) on page 87](#)
- [SRC CLI Commands to Monitor the SRC Diameter Server on page 97](#)

## Viewing SRC Diameter Server State (SRC CLI)

---

**Purpose** View status information about the Diameter server.

**Action** To display information about the status 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
```

**Related  
Documentation**

- [Configuring the Diameter Application \(SRC CLI\) on page 87](#)
- [Configuring Diameter Peers \(SRC CLI\) on page 93](#)
- [SRC CLI Commands to Monitor the SRC Diameter Server on page 97](#)

## CHAPTER 12

# Managing Services with Diameter on MX Series Routers

- [SRC Peer Support on MX Series Routers Overview on page 101](#)
- [Managing Services on MX Series Routers Using the Diameter Application on page 102](#)
- [Configuring JSRC on the MX Series Router on page 103](#)
- [Configuring the Diameter Application \(SRC CLI\) on page 103](#)
- [Adding Network Devices \(SRC CLI\) on page 109](#)
- [Configuring Diameter Peers \(SRC CLI\) on page 111](#)
- [Configuring the SAE to Manage Network Devices \(SRC CLI\) on page 113](#)
- [Specifying Initialization Scripts for the Intelligent-Service-Edge Device Driver \(SRC CLI\) on page 117](#)
- [Configuring JSRC Policies \(SRC CLI\) on page 118](#)

### **SRC Peer Support on MX Series Routers Overview**

---

When the Juniper Networks routing platform supports the use of the Diameter protocol to provide extended authentication, authorization, and accounting (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 service activation engine (SAE). The local SRC peer is known as Junos OS (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  
Documentation**

- [Managing Services on MX Series Routers Using the Diameter Application on page 102](#)
- [Configuring JSRC on the MX Series Router on page 103](#)
- [Configuring the SAE to Manage Network Devices \(SRC CLI\) on page 113](#)
- [Configuring Diameter Peers \(SRC CLI\) on page 93](#)

---

## 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 103](#)
- [Configuring the Diameter Application \(SRC CLI\) on page 87](#)
- [Adding Network Devices \(SRC CLI\) on page 109](#)
- [Configuring Diameter Peers \(SRC CLI\) on page 93](#)
- [Configuring the SAE to Manage Network Devices \(SRC CLI\) on page 113](#)
- [Configuring JSRC Policies \(SRC CLI\) on page 118](#)



## Configuring JSRC on the MX Series Router

Tasks to set up JSRC on the Juniper Networks routing platform are:

1. Configure the Diameter instance.  
See [“Configuring the Diameter Application \(SRC CLI\)” on page 87](#).
2. Set up the MX Series router so that it can be managed by the SAE.  
See [“Adding Network Devices \(SRC CLI\)” on page 109](#).
3. Configure the Diameter peer.  
See [“Configuring Diameter Peers \(SRC CLI\)” on page 93](#).
4. Configure the SAE to manage the MX Series router.  
See [“Configuring the SAE to Manage Network Devices \(SRC CLI\)” on page 113](#).
5. Configure JSRC policies.  
See [“Configuring JSRC Policies \(SRC CLI\)” on page 118](#).

For more information about JSRC and subscriber access, see the *Junos OS Broadband Subscriber Management and Services Library*.

## Configuring the Diameter Application (SRC CLI)

You can configure the properties of the application, client, server, and logging destination of the SRC Diameter application.

Perform the following tasks to configure these properties:

- [Configuring the Diameter Application Properties on page 103](#)
- [Configuring the Diameter Client Properties on page 107](#)
- [Configuring the Diameter Server Properties on page 108](#)
- [Configuring Logging Destinations on page 108](#)

## Configuring the Diameter Application Properties

The SRC software supports Diameter application properties such as Juniper Networks Session Resource Control (JSRC) and southbound Gx interface. JSRC and southbound Gx interface communicate with the Service Activation Engine (SAE) (remote SRC peer).

Use the following configuration statements to configure the properties for the Diameter application:

```
system diameter {
  java-heap-size java-heap-size;
  java-new-size java-new-size;
  java-garbage-collection-options java-garbage-collection-options;
  protocol [(tcp | sctp)...];
  local-address [local-address...];
```

```
port port;  
origin-host origin-host;  
origin-realm origin-realm;  
diameter-server-timeout diameter-server-timeout;  
active-peers;  
debug-mode;  
load-balancing-mode (failover | round-robin);  
transaction-processing-log (log-no-messages | log-severe-messages |  
    log-normal-messages | log-debug-messages);  
packet-trace-log (log-no-messages | log-severe-messages | log-normal-messages |  
    log-debug-messages);  
peer-state-machine-log (log-no-messages | log-severe-messages | log-normal-messages  
    | log-debug-messages);  
configuration-log (log-no-messages | log-severe-messages | log-normal-messages |  
    log-debug-messages);  
}
```

To configure the Diameter application:

1. From configuration mode, access the statement for the Diameter application.

```
user@host# edit system diameter
```



**NOTE:** The java-\* options have default values that should not be changed unless directed by Juniper Networks Technical Assistance Center (JTAC).

2. If you encounter problems caused by lack of memory, change the maximum memory size available to the Java Runtime Environment (JRE).

```
[edit system diameter]  
user@host# set java-heap-size java-heap-size
```

3. Configure the amount of space available to the JRE when the Diameter server starts.

```
[edit system diameter]  
user@host# set java-new-size java-new-size
```

4. Configure the garbage collection functionality of the Java Virtual Machine.

```
[edit system diameter]  
user@host# set java-garbage-collection-options java-garbage-collection-options
```

5. Specify the protocol for the transport connection.

```
[edit system diameter]  
user@host# set protocol [(tcp | sctp) ...]
```

6. (Optional) Specify the local IP addresses that remote peers can use to reach this server.

```
[edit system diameter]
user@host# set local-address [local-address...]
```

7. (Optional) Specify the port for the server.

```
[edit system diameter]
user@host# set port port
```

8. (Optional) Specify the fully qualified domain name (FQDN) used to identify this host to its Diameter peers.

```
[edit system diameter]
user@host# set origin-host origin-host
```

9. (Optional) Specify the realm used to identify this host to its Diameter peers.

```
[edit system diameter]
user@host# set origin-realm origin-realm
```

The Diameter realm should be configured to the domain name of the origin host. For example, if the FQDN of the host is host.juniper.net, then the realm should be juniper.net.

10. (Optional) Configure the timeout value until which the Diameter server holds unsolicited requests such as Point to Point Protocol (PPP) and Abort Session Request (ASR), and waits for a matching response such as Push Profile Answer (PPA) and Abort Session Answer (ASA). The server discards the responses received after the specified time. The value range is 1–65,565 seconds. The preferred value is 10–30 seconds. By default, the value is set to 25 seconds.

```
[edit system diameter]
user@host# set diameter-server-timeout diameter-server-timeout
```



**NOTE:** `diameter-server-timeout` and `reply-timeout` under the `[edit shared sae group configuration driver]` hierarchy should be configured with the same value.

11. (Optional) Specify whether the peer connection is in active mode.

```
[edit system diameter]
user@host# set active-peers
```



---

**NOTE:**

- Active mode means that the SRC software actively tries to connect to the peer. Make sure the peer you are connecting to supports active peers. The MX Series router does not support active peers. The SRC software can still be configured, but the connection attempts will not work.
  - If the peer connection is configured to be in active mode, you must configure the remote peer address for all Diameter peers by using the **address** option under the **[edit shared network diameter peer *name*]** hierarchy.
- 

12. (Optional) Specify whether the peer connection is in debug mode.

```
[edit system diameter]
user@host# set debug-mode
```

13. (Optional) Configure the load-balancing mode for peer selection when forwarding a request message.

```
[edit system diameter]
user@host# set load-balancing-mode (failover | round-robin)
```

14. (Optional) Configure the log level for the transaction processing log.

```
[edit system diameter]
user@host# set transaction-processing-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

15. (Optional) Configure the log level for the packet tracing log.

```
[edit system diameter]
user@host# set packet-trace-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

16. (Optional) Configure the log level for the peer state machine log.

```
[edit system diameter]
user@host# set peer-state-machine-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

17. (Optional) Configure the log level for the configuration log.

```
[edit system diameter]
user@host# set configuration-log log-level
```

where *log-level* is one of the following:

- **log-no-messages**—Do not log any messages.
- **log-severe-messages**—Log only severe messages.
- **log-normal-messages**—Log only normal messages.
- **log-debug-messages**—Log only debug messages.

## Configuring the Diameter Client Properties

This procedure configures the client-side adapter of the SRC Diameter server, which handles client connections. Configuration should be necessary only if you encounter performance problems.

Use the following statements to configure the properties for the Diameter client:

```
system diameter client {
  threads threads;
  keep-alive-time keep-alive-time;
}
```

To configure the Diameter client properties:

1. From configuration mode, access the statement for the Diameter client.

```
user@host# edit system diameter client
```

2. (Optional) Specify the number of threads to use.

```
[edit system diameter client]
user@host# set threads threads
```

3. (Optional) Specify the time to wait for new commands.

```
[edit system diameter client]
user@host# set keep-alive-time keep-alive-time
```

- See Also**
- [Configuring the Diameter Server Properties on page 92](#)
  - [Configuring Logging Destinations on page 92](#)

## 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
```

- See Also**
- [Configuring the Diameter Client Properties on page 91](#)
  - [Configuring Logging Destinations on page 92](#)

## Configuring Logging Destinations

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

```
system diameter logger name ...
```

```
system diameter logger name file {  
  filter filter;  
  filename filename;  
  rollover-filename rollover-filename;  
  maximum-file-size maximum-file-size;  
}
```

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

1. From configuration mode, access the statement that configures the name and type of logging destination.

```
user@host# edit system diameter logger name file
```

2. Specify the properties for the logging destination.

```
[edit system diameter logger name file]  
user@host# set ?
```

For more information about configuring properties for the logging destination, see *Configuring Logging Destinations to Store Messages in a File (SRC CLI)*.

- Related Documentation**
- [SRC CLI Commands to Monitor the SRC Diameter Server on page 97](#)
  - To manage services for JSRC peers on MX Series routers, see [Managing Services on MX Series Routers Using the Diameter Application on page 102](#).

---

## 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. 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. (Optional) Specify the origin hostname. This example procedure uses `mx1-origin-host` as the origin hostname. If the origin hostname is not configured, SAE uses the device name (`mx1` in the example) as the origin hostname. If configured, the mentioned origin hostname must match the origin hostname of the Diameter peer (for example, MX Series router).

```
[edit shared network device mx1]
user@host# set origin-host mx1-origin-host
```



**NOTE:** If the origin hostname is configured under the [edit shared network device *name*] hierarchy, the device name does not need to be same as the origin hostname of the Diameter peer. Otherwise, the device name must match the origin hostname of the Diameter peer.

4. Specify the configured peers associated with the device. See [“Configuring Diameter Peers \(SRC CLI\)” on page 93](#).

```
[edit shared network device mx1]
user@host# set peers [peers...]
```



**NOTE:** MX Series routers support only a single peer connection.

5. 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.

6. Specify the SAEs that can manage this router.

```
[edit shared network device mx1 virtual-router default]
user@host# set sae-connection [sae-connection...]
```

7. (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)
```

8. (Optional) Verify your configuration.

```
[edit shared network device mx1]
user@host# show
device-type junos-ise;
origin-host mx1-origin-host;
peers bng-srcmx480b;
virtual-router * {
  sae-connection 10.212.10.2;
```



```
vpn-id 123;
}
```

**Related  
Documentation**

- [Configuring the SAE to Manage Network Devices \(SRC CLI\) on page 113](#)
- [Configuring JSRC on the MX Series Router on page 103](#)

## Configuring Diameter Peers (SRC CLI)

Use the following configuration statements to configure the Diameter peers:

```
shared network diameter peer name {
  protocol [(tcp | sctp)...];
  address [address...];
  enforce-source-address;
  local-address local-address;
  connect-timeout connect-timeout;
  watchdog-timeout watchdog-timeout;
  state-machine-timeout state-machine-timeout;
  reconnect-timeout reconnect-timeout;
  port port;
  origin-host origin-host;
  incoming-queue-limit incoming-queue-limit;
  active-peer;
}
```



**NOTE:** When you commit the Diameter peer configuration, keep in mind the following conditions:

- The origin host, remote peer address, or both should be specified for the Diameter peer.
- If the enforce source address is configured for the Diameter peer, the remote peer address should be specified for the Diameter peer.
- If the peer connection is configured to be in active mode for a particular Diameter peer or globally for all Diameter peers by using the `active-peers` option under the `[edit system diameter]` hierarchy, the remote peer address should be specified for the Diameter peers.

To configure the Diameter peer:

1. From configuration mode, access the statements for the peer.

```
user@host# edit shared network diameter peer name
```

The peer name must be unique.

2. Specify the protocol for the transport connection.

```
[edit shared network diameter peer name]  
user@host# set protocol [(tcp | sctp)...
```

3. (Optional) Specify the addresses of the remote peer. If SCTP is the transport protocol, you can specify multiple addresses. If TCP is the transport protocol, you can specify only a single address.

```
[edit shared network diameter peer name]  
user@host# set address [address...
```

4. (Optional) Specify whether the remote peer must connect from one of the IP addresses listed by the **address** option.

```
[edit shared network diameter peer name]  
user@host# set enforce-source-address
```

5. (Optional) Specify the local address of the peer.

```
[edit shared network diameter peer name]  
user@host# set local-address local-address
```

6. (Optional) Specify the maximum amount of time allowed for the Diameter peer to respond to a connection request.

```
[edit shared network diameter peer name]  
user@host# set connect-timeout connect-timeout
```

7. (Optional) Specify the watchdog timeout used for the connection to the remote peer.

```
[edit shared network diameter peer name]  
user@host# set watchdog-timeout watchdog-timeout
```

8. (Optional) Specify the Diameter state machine timeout.

```
[edit shared network diameter peer name]  
user@host# set state-machine-timeout state-machine-timeout
```

9. (Optional) Specify the time interval between connection attempts when the peer is in the disconnected state.

```
[edit shared network diameter peer name]  
user@host# set reconnect-timeout reconnect-timeout
```

10. (Optional) Specify the port for the client.

```
[edit shared network diameter peer name]
user@host# set port port
```

11. (Optional) Specify the identifier for the endpoint that the peer presents during connection establishment.

```
[edit shared network diameter peer name]
user@host# set origin-host origin-host
```

12. (Optional) Specify the number of messages allowed on the incoming message queue for a peer.

```
[edit shared network diameter peer name]
user@host# set incoming-queue-limit incoming-queue-limit
```

13. (Optional) Specify whether the peer connection is in active mode.

```
[edit shared network diameter peer name]
user@host# set active-peer
```



**NOTE:** Active mode means that the SRC software actively tries to connect to the peer. Make sure the peer you are connecting to supports active peers. The MX Series router does not support active peers. The SRC software can still be configured, but the connection attempts will not work.

#### Related Documentation

- [Configuring the Diameter Application \(SRC CLI\) on page 87](#)
- [Viewing SRC Diameter Server State \(SRC CLI\) on page 99](#)

## 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;
  pool-retrieval;
  sync-from-sessionstore;
  ignore-framed-ip-netmask;
  cached-driver-expiration cached-driver-expiration;
  concurrent-post-sync-jobs concurrent-post-sync-jobs;
  concurrent-request-timeout concurrent-request-timeout;
  concurrent-requests concurrent-requests ;
  enable-disconnect-ontimeout;
```

```
delay-service-policy-provisioning delay-service-policy-provisioning
keep-alive-timeout keep-alive-timeout;
pending-acrs-strs-wait-time pending-acrs-strs-wait-time;
registry-retry-interval registry-retry-interval;
reply-timeout reply-timeout;
sequential-message-timeout sequential-message-timeout;
sync-count-wait-timeout sync-count-wait-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 pool retrieval option.

```
[edit shared sae configuration driver junos-ise]
user@host# set pool-retrieval
```

4. (Optional) Specify whether the SAE should be synchronized from the session store.

```
[edit shared sae group POP-ID configuration driver junos-ise]
user@host# set sync-from-sessionstore
```

5. (Optional) Specify whether to ignore the Framed-IP-Mask AVP and allow IP-based filtering without considering the framed IP netmask.

```
[edit shared sae group POP-ID configuration driver junos-ise]
user@host# set ignore-framed-ip-netmask
```

6. (Optional) Specify the number of jobs that can be processed concurrently to log in to subscriber sessions that are incomplete after synchronizing state with the router. You can configure a value ranging from 10 through 50. Default value is 20.

```
[edit shared sae configuration driver junos-ise]
user@host# set concurrent-post-sync-jobs concurrent-post-sync-jobs
```

7. (Optional) Specify the timeout for sending concurrent requests. You can configure a value ranging from 0 through 900 seconds. Default value is 30 seconds.

```
[edit shared sae configuration driver junos-ise]  
user@host# set concurrent-request-timeout concurrent-request-timeout
```

8. (Optional) Specify the number of unsolicited requests that can be sent concurrently. You can configure a value ranging from 1 through 500. Default value is 100.

```
[edit shared sae configuration driver junos-ise]  
user@host# set concurrent-requests concurrent-requests
```

9. (Optional) Specify whether the user session needs to be removed from the router.

```
[edit shared sae configuration driver junos-ise]  
user@host# set enable-disconnect-ontimeout
```

10. (Optional) Specify the amount of time by which scheduler tasks are delayed after the user login is completed. You can configure a value ranging from 0 through 1000 milliseconds. Default value is 0.

```
[edit shared sae configuration driver junos-ise]  
user@host# set delay-service-policy-provisioning delay-service-policy-provisioning
```

11. (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
```

12. (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
```

13. (Optional) Specify the maximum time that the device driver waits for completing sessions restoration to start processing pending Accounting-Request (ACR) and Session-Termination-Request (STR) messages. You can configure a value ranging from 600 through 18000 seconds. Default value is 3600 seconds.

```
[edit shared sae configuration driver junos-ise]  
user@host# set pending-acrs-strs-wait-time pending-acrs-strs-wait-time
```

14. (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
```

15. (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
```

16. (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
```

17. (Optional) Specify the interval after which SAE stops waiting for the sync-AAR messages and triggers unsolicited synchronization. You can configure a value ranging from 0 through 20132147483647 seconds. Default value is 2 seconds.

```
[edit shared sae configuration driver junos-ise]
user@host# set sync-count-wait-timeout sync-count-wait-timeout
```

18. (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
```

19. (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
```

20. (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)*.

**Related  
Documentation**

- [Adding Network Devices \(SRC CLI\) on page 109](#)
- [Configuring the Diameter Application \(SRC CLI\) on page 87](#)
- [Configuring Local Properties for the SAE \(SRC CLI\)](#)
- [SRC Peer Support on MX Series Routers Overview on page 101](#)

## Specifying Initialization Scripts for the Intelligent-Service-Edge Device Driver (SRC CLI)

Use the following configuration statements to specify initialization scripts for the intelligent-service-edge device driver:

```
shared sae configuration driver scripts {
  extension-path extension-path;
  general general;
  junos-ise junos-ise;
}
```

To configure initialization scripts for the intelligent-service-edge device driver:

1. From configuration mode, access the configuration statements that configure initialization scripts. In this sample procedure, the scripts are configured in the west-region group.

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

2. Specify the initialization script for the intelligent-service-edge device driver.

```
[edit shared sae group west-region configuration driver scripts]
user@host# set junos-ise junos-ise
```

SAE runs the specified script when the intelligent-service-edge device driver is activated and again when the driver is deactivated.

3. Configure the initialization script that can be used for all other types of routers supported by the SRC module.

```
[edit shared sae group west-region configuration driver scripts]
user@host# set general general
```

4. Configure a path to initialization scripts that are not in the default location, `/opt/UMC/sae/lib`.

```
[edit shared sae group west-region configuration driver scripts]
user@host# set extension-path extension-path
```

5. (Optional) From operational mode, verify your initialization script configuration.

```
[edit shared sae group west-region configuration driver scripts]
user@host# show
junos-ise isePoolPublisher;
```

- Related Documentation**
- [Copying Initialization Scripts to the C Series Controller](#)
  - [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers](#)

## Configuring JSRC Policies (SRC CLI)

---

Tasks to configure JSRC policies are:

- [Configuring JSRC Policy Lists on page 118](#)
- [Configuring JSRC Policy Rules on page 118](#)
- [Configuring Dynamic Profile Actions on page 119](#)
- [Configuring Operation Script for Policy Provisioning \(SRC CLI\) on page 120](#)

### 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
```

For more information about dynamic profiles and subscriber access, see the *Junos OS Broadband Subscriber Management and Services Library*.

**See Also** • [Configuring JSRC on the MX Series Router on page 103](#)

## Configuring Operation Script for Policy Provisioning (SRC CLI)

You can use operation scripts to support the policy provisioning for JSRC policy rules. The SRC software passes the operation script values configured by using the **operation-script** option under the **[edit policies group *name* list *name* rule *name*]** hierarchy level to the Extensible Subscriber Services Manager Daemon on the MX Series router. You can assign the operation script only to the rules for which the role of the policy list is set as **junos-ise** and the **applicability** is set as **both**.



### NOTE:

- AA-Answer message can have both dynamic profile and operation script in the policy rule, whereas the Push-Profile-Request can have either dynamic profile or operation script in the policy rule.
  - In the policy rule configuration, the **dynamic-profile** and **operation-script** options are mutually exclusive.
- 

Use the following configuration statements to configure an operation script for JSRC policy rules:

```
policies group name list name rule name operation-script{  
  description description;  
  script-name script-name;  
  script-args-format script-args-format ;  
}
```

```

}
policies group name list name rule name operation-script variables name {
    value value;
    type type;
}

```

To configure an operation script for JSRC policy rules:

1. From configuration mode, enter the operation script configuration.

```

[edit policies group name list name rule name]
user@host# set operation-script

```

2. (Optional) Enter a description for the operation script.

```

[edit policies group name list name rule name operation-script]
user@host# set description description

```

3. Enter a name for the operation script.

```

[edit policies group name list name rule name operation-script]
user@host# set script-name script-name

```

4. Enter the operation script arguments.

```

[edit policies group name list name rule name operation-script]
user@host# set script-args-format script-args-format

```

Use the format '*`\${arg1}`;`\${arg2}`;`\${arg3}`*'.

For example: '*`\${user\_ipAddress}`;[vlan]*';



#### NOTE:

- You must enclose the arguments in quotation marks.
- The operation script argument name must match a variable name configured for policy provisioning.

5. From configuration mode, enter the parameters used by the operation script for policy provisioning.

```

[edit]
user@host# set policies group name list name rule name operation-script variables
name]

```

6. (Optional) Configure a value for the variable.

```

[edit policies group name list name rule name operation-script variables name]

```

```
user@host# set value value
```

7. (Optional) Configure the variable type. Variable types are mapped to parameter types.

```
[edit policies group name list name rule name operation-script variables name]  
user@host# set type type
```

8. (Optional) Verify the operation script configuration.

```
[edit policies group name list name rule name  
user@host# show  
operation-script {  
  script-args-format '[user_ipAddress];[vlan]';  
  script-name ngcoco;  
  variables {  
    var1 {  
      type any;  
      value user_ipAddress;  
    }  
    var2 {  
      type any;  
      value vlan;  
    }  
  }  
}  
type junos-ise;
```

- See Also**
- *Configuring Dynamic Profile Actions (SRC CLI)*
  - *Policy Rules Overview*

- Related Documentation**
- [Configuring JSRC on the MX Series Router on page 103](#)
  - *Policy Rules Overview*

## CHAPTER 13

# Managing Subscriber Sessions on MX Series Routers in an SRC Network

- [Subscriber Sessions on MX Series Routers Overview on page 123](#)
- [Managing Subscriber Sessions on MX Series Routers \(SRC CLI\) on page 124](#)
- [Viewing Statistics for the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 141](#)
- [Monitoring Statistics for the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 142](#)

## Subscriber Sessions on MX Series Routers Overview

---

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 virtual local area network (VLAN).
- One address subscriber session for each Dynamic Host Configuration Protocol (DHCP) address.

You can manage subscriber sessions with the External Subscriber Monitor application and the change-of-authorization (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.

### Related Documentation

- [Configuring External Subscriber Monitor \(SRC CLI\) on page 124](#)
- [Setting Up MX Series Routers in the SRC Network \(SRC CLI\) on page 137](#)

## 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 \(SRC CLI\) on page 124](#)
- [Configuring Pseudo–RADIUS Authorization Server Properties \(SRC CLI\) on page 125](#)
- [Configuring the NIC Proxy for the Pseudo–RADIUS Authorization Server \(SRC CLI\) on page 130](#)
- [Extracting RADIUS Attributes with the Pseudo–RADIUS Authorization Server \(SRC CLI\) on page 134](#)
- [Enabling the Pseudo–RADIUS Authorization Server \(SRC CLI\) on page 137](#)
- [Disabling the Pseudo–RADIUS Authorization Server \(SRC CLI\) on page 137](#)
- [Setting Up MX Series Routers in the SRC Network \(SRC CLI\) on page 137](#)
- [Configuring the COA Script Service for MX Series Routers \(SRC CLI\) on page 138](#)
- [Configuring Parameters for the Script Service for MX Series Routers \(SRC CLI\) on page 139](#)
- [Configuring Subscriptions to the Script Service on page 141](#)

### 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)*.

**See Also** • [Configuring Pseudo–RADIUS Authorization Server Properties \(SRC CLI\) on page 125](#)

- [Configuring the NIC Proxy for the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 130](#)
- [Extracting RADIUS Attributes with the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 134](#)
- [Setting Up MX Series Routers in the SRC Network \(SRC CLI\) on page 137](#)

## Configuring Pseudo-RADIUS Authorization Server Properties (SRC CLI)

Tasks to configure the pseudo-RADIUS authorization server are:

- [Configuring the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 125](#)
- [Configuring the Directory Connection Properties for the Subscriber Data on page 128](#)
- [Configuring Directory Connection Properties for the Cached DHCP Profiles on page 129](#)

### 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.

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
```

- See Also**
- [Configuring External Subscriber Monitor \(SRC CLI\) on page 124](#)
  - [Configuring the NIC Proxy for the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 130](#)
  - [Extracting RADIUS Attributes with the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 134](#)
  - [Enabling the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 137](#)
  - [Viewing Statistics for the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 141](#)
  - [Monitoring Statistics for the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 142](#)

## 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 131](#)
- [Changing the Configuration for the NIC Proxy Cache on page 131](#)
- [Configuring a NIC Proxy for NIC Replication on page 132](#)

---

### 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 Normal 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;
```

- See Also**
- [Configuring External Subscriber Monitor \(SRC CLI\) on page 124](#)
  - [Configuring Pseudo-RADIUS Authorization Server Properties \(SRC CLI\) on page 125](#)
  - [Extracting RADIUS Attributes with the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 134](#)
  - [Enabling the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 137](#)

## 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 134](#)
- [Extracting Virtual Router Name Attribute Values on page 135](#)

### 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://docs.oracle.com/javase/1.5.0/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://docs.oracle.com/javase/1.5.0/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
```

- See Also**
- [Configuring External Subscriber Monitor \(SRC CLI\) on page 124](#)
  - [Configuring Pseudo-RADIUS Authorization Server Properties \(SRC CLI\) on page 125](#)

- [Configuring the NIC Proxy for the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 130](#)
- [Enabling the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 137](#)

### 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
```

- See Also**
- [Configuring External Subscriber Monitor \(SRC CLI\) on page 124](#)
  - [Configuring Pseudo-RADIUS Authorization Server Properties \(SRC CLI\) on page 125](#)
  - [Disabling the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 137](#)
  - [Viewing Statistics for the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 141](#)
  - [Monitoring Statistics for the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 142](#)

### 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
```

- See Also**
- [Enabling the Pseudo-RADIUS Authorization Server \(SRC CLI\) on page 137](#)

### 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...]
```

- See Also**
- [Configuring External Subscriber Monitor \(SRC CLI\) on page 124](#)
  - [Configuring Pseudo–RADIUS Authorization Server Properties \(SRC CLI\) on page 125](#)
  - [Configuring the COA Script Service for MX Series Routers \(SRC CLI\) on page 138](#)
  - [Configuring Subscriptions to the Script Service on page 141](#)

## 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.smgmt.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 website at:

<https://www.juniper.net/support/downloads/?p=src#sw>

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.smgmt.scriptServices.coa.CoaService;
  file file:///opt/UMC/sae/lib/coa.jar;
}
```

6. Configure the parameters for the script service.

See “Configuring Parameters for the Script Service for MX Series Routers (SRC CLI)” on page 139.

- See Also**
- [Setting Up MX Series Routers in the SRC Network \(SRC CLI\) on page 137](#)
  - [Configuring Parameters for the Script Service for MX Series Routers \(SRC CLI\) on page 139](#)
  - [Configuring Subscriptions to the Script Service on page 141](#)

## 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 140 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	<p>Content of service definition in the format</p> <pre>&lt;action&gt;.&lt;radiusAttributeName&gt;=&lt;pluginEventAttribute&gt;\n</pre> <ul style="list-style-type: none"> <li>• <b>action</b>—Action that is executed on packet content (attribute): <ul style="list-style-type: none"> <li>• start</li> <li>• stop</li> <li>• start-stop</li> </ul> </li> <li>• <b>radiusAttributeName</b>—Valid RADIUS attribute specified as follows: <ul style="list-style-type: none"> <li>• Standard RADIUS attribute name or number</li> <li>• VSA in the format vendor-specific.&lt;vendor#&gt;.&lt;vsa#&gt;.string</li> </ul> </li> <li>• <b>pluginEventAttribute</b>—Valid Python expression</li> <li>• <b>\n</b>—New-line character included between the lines of a configuration containing multiple lines; the entire configuration must be enclosed in quotation marks.</li> </ul> <p>For example:</p> <pre>start-stop.Acct-Session-Id = ifSessionId</pre> <pre>"start-stop.Acct-Session-Id=ifSessionId\nstart.vendor-specific.4874.10.string='video'\nstop.vendor-specific.4874.10.string='default'\n"</pre>

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=['T011m', '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
```

- See Also**
- [Configuring the COA Script Service for MX Series Routers \(SRC CLI\) on page 138](#)
  - [Configuring Subscriptions to the Script Service on page 141](#)

## 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)*.

- See Also**
- [Configuring the COA Script Service for MX Series Routers \(SRC CLI\) on page 138](#)
  - [Configuring Parameters for the Script Service for MX Series Routers \(SRC CLI\) on page 139](#)

## 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
```

- Related Documentation**
- [Configuring Pseudo–RADIUS Authorization Server Properties \(SRC CLI\) on page 125](#)
  - [Monitoring Statistics for the Pseudo–RADIUS Authorization Server \(SRC CLI\) on page 142](#)

## 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
```

**Related Documentation**

- [Configuring Pseudo–RADIUS Authorization Server Properties \(SRC CLI\) on page 125](#)
- [Viewing Statistics for the Pseudo–RADIUS Authorization Server \(SRC CLI\) on page 141](#)



# Configuring Services for SRC-Managed Routers

- [DPI Script Service Overview on page 143](#)
- [Creating the DPI Script Service \(SRC CLI\) on page 144](#)
- [Configuring Subscriptions to the DPI Script Service on page 145](#)
- [Parameters for DPI Script Service on page 147](#)
- [Creating a Configuration File on page 149](#)
- [Example: Using the DPI Script Service on page 156](#)

## DPI Script Service Overview

---

The SRC software has a Deep Packet Inspection (DPI) script service that can be used to modify services on routers managed by the SRC software to provide resource management based on supported applications. The DPI script service allows the SRC software to manage services on Juniper Networks routers running Junos OS, such as the Juniper Networks MX Series Ethernet Services Router. When the service activation engine (SAE) activates the DPI script service, the session uses the command channel to manage policies on MX Series routers. The DPI script service can use the Junos XML management protocol to change the configuration on the MX Series router as well as send commands to third-party devices that support Telnet access.

The DPI script service activates services by applying policy rules to the managed interface on a network device. The DPI script service allows you to modify existing rules by parameter substitutions; that is, you can replace existing rules with new rules or delete existing rules. The DPI script service supports only default SRC subscriptions and default SAE service sessions. The script service supports policy rules to handle Junos XML management protocol requests for MX Series routers and Telnet commands for third-party routers.

### Related Documentation

- For information about the Junos XML management protocol, see the *Junos XML Management Protocol Developer Guide*
- For information about service set configuration on routers running Junos OS, see the *Junos OS Services Interfaces Library for Routing Devices*

- For information about accessing and configuring the third-party device using Telnet, see the device's software documentation

## Creating the DPI Script Service (SRC CLI)

---

To create the script service:

1. Create a script service in the [**edit services scope *name* service *name***] hierarchy. In this sample procedure, the service is configured in the DPI service scope, and DPI is the name of the service.

```
user@host# edit services scope DPI service DPI
```

2. Set the type of service to script.

```
[edit services scope DPI service DPI]  
user@host# set type script
```

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

```
[edit services scope DPI service DPI]  
user@host# edit script
```

5. Configure *net.juniper.smgtpiscriptservice.DpiService* as the name of the class that implements the script service.

```
[edit services scope DPI service DPI script]  
user@host# set class-name net.juniper.smgtpiscriptservice.DpiScriptService
```

6. Configure Java archive file as the type of script that the script service uses.

```
[edit services scope DPI service DPI script]  
user@host# set script-type java-archive
```

7. Specify the filename of the script service implementation so that its contents will be loaded into the **file** option. Copy the *dpiss.jar* file to the C Series Controller before you specify the filename. In this sample procedure, the *dpiss.jar* file was copied from a location that is accessible by a URL (such as an FTP or HTTP server) to the */tmp* directory.

```
[edit services scope DPI service DPI script]  
user@host# run file copy URL /tmp/dpiss.jar  
user@host# set filename /tmp/dpiss.jar
```

8. (Optional) From configuration mode, enter the service parameter configuration and configure values for parameters.

```
user@host# edit services scope DPI service DPI parameter
```

```
[edit services scope DPI service DPI parameter]
```

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

For example, to specify the configuration file that is used or to specify whether the configuration file is reloaded for each script service activation:

```
user@host# set substitution dpiConfig="\resource/dpiConf.xml\"
```

```
user@host# set substitution dpiConfigDebug="\off\"
```

9. (Optional) Verify your configuration.

```
[edit services scope DPI service DPI]
```

```
user@host# show
```

**Related Documentation**

- *SRC Script Services Overview*

## Configuring Subscriptions to the DPI Script Service

You configure subscriptions to the DPI script service by adding subscribers. You can set up the subscription to activate immediately on login.

To add access subscribers:

1. From configuration mode, enter the `[edit subscribers retailer]` hierarchy. In this sample procedure, the retailer called DPI is configured in the DPI service scope.

```
user@host# edit subscribers retailer DPI
```

2. Specify the domain name associated with the retailer.

```
[edit subscribers retailer DPI]
```

```
user@host# set domain-name [domain-name...]
```

3. (Optional) Assign service scopes for the retailer.

```
[edit subscribers retailer DPI]
```

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

4. Add a subscriber folder for the retailer. In this sample procedure, local is the name of the subscriber folder.

```
[edit subscribers retailer DPI]
user@host# edit subscriber-folder local
```

5. Add an enterprise subscriber. In this sample procedure, ENT is the name of the enterprise subscriber.

```
[edit subscribers retailer DPI subscriber-folder local]
user@host# edit enterprise ENT
```

6. (Optional) Configure values for parameters.

```
[edit subscribers retailer DPI subscriber-folder local enterprise ENT]
user@host# set substitution [substitution...]
```

For example, to specify the dpiRules or dpiAdminState parameter:

```
user@host# set substitution 'dpiRules=[{app="rtsp", action="accept",
    fcl="expedited-forwarding"}, {app="bittorrent", action="discard"}]'
user@host# set substitution dpiAdminState=\"enabled\"
```

7. Configure accesses for the enterprise subscriber.

```
[edit subscribers retailer DPI subscriber-folder local enterprise ENT]
user@host# edit access name
```

8. Specify the interface name associated with the access using the CLI syntax of the device.

```
[edit subscribers retailer DPI subscriber-folder local enterprise ENT access name]
user@host# set interface-name interface-name
```

9. (Optional) Configure actual values for parameters.

```
[edit subscribers retailer DPI subscriber-folder local enterprise ENT access name]
user@host# set substitution [substitution...]
```

For example, to specify the interface class:

```
user@host# set substitution 'dpiInterfaceClasses=["MXEnterprise"]'
```

10. From configuration mode, enter the subscription configuration. In this sample procedure, the service called DPI is configured for the enterprise.

```
user@host# edit subscribers retailer DPI subscriber-folder local enterprise ENT
subscription DPI
```

11. Specify that the service is activated on login.

```
[edit subscribers retailer DPI subscriber-folder local enterprise ENT subscription DPI]
user@host# set activation automatically-on-login
```

#### Related Documentation

- *Adding Subscribers (SRC CLI)*
- *Adding Retailers (SRC CLI)*
- *Adding Subscriber Folders (SRC CLI)*
- *Adding Enterprises (SRC CLI)*
- *Configuring Accesses (SRC CLI)*
- *Configuring Subscriptions (SRC CLI)*
- *Configuring Subscribers and Subscriptions Overview*

## Parameters for DPI Script Service

Table 10 on page 148 lists the parameters specified by the DPI script service, which is implemented by the `/SDK/scriptServices/dpiScriptService/lib/dpiss.jar` file found in the `SDK+AppSupport+Demos+Samples.tar.gz` file. The value assigned to the parameter must be enclosed by quotation marks. For example, `dpiConfig="resource/dpiConf.xml"` specifies the pathname for the configuration file.

Table 10: Parameter Substitutions for DPI Services

Parameter Name	Description
dpiConfig	<p>Configuration file in the format of a URL or pathname. The script service handles each format as follows:</p> <ul style="list-style-type: none"> <li>URL—String that starts with <b>http</b>:. The script service uses the value to download the configuration file.</li> <li>pathname—String that does not start with <b>http</b>:. The script service uses the value as a path to a resource in the <i>.jar</i> file that contains the script service.</li> </ul> <p>We recommend using a pathname in a production environment.</p> <p>If you do not supply a value, the default value is <b>"resource/dpiConf.xml"</b>.</p>
dpiConfigDebug	<p>Reloads the configuration file specified by <b>dpiConfig</b> for each script service activation. Specify <b>"on"</b> to perform the reload. The default value is <b>"off"</b>, where the script service uses the configuration file that is accessed for the first service activation until the SAE is restarted.</p>
dpiAdminState	<p>Used by the application to activate the service. By default, the value is set to <b>"enabled"</b>. Set the value to <b>"disabled"</b> to deactivate the service.</p> <p><b>NOTE:</b> If you set the value to disabled, the service session continues to exist because the service is not deactivated. This behavior allows the application to monitor the operational state of the service so that the application knows when a configuration change has committed.</p>
dpiOprState	<p>Used by the application to determine the operational state of the service.</p> <p>When you activate, deactivate, or change the parameters of the DPI script service, the operational state is set to <b>commit pending</b>. Once the commit succeeds, the operational state changes to <b>committed</b>.</p> <p><b>NOTE:</b> The dpiOprState parameter is reserved for the use of the DPI script service. Another application, such as an enterprise Web application, cannot use this parameter.</p>
dpiMaintMode	<p>Prevents the script service from making configuration changes. You can set this parameter in a service scope attached to a specific router so that it affects only that router. If you set this parameter in the service, the whole network is put in maintenance mode.</p> <p>Specify <b>"off"</b> (the default value) to turn off maintenance mode.</p> <p>You can specify <b>"on"</b> to allow the script service to continue recording configuration changes and to maintain the corresponding service sessions in the commit pending state until maintenance mode is set to <b>"off"</b>.</p>

Table 10: Parameter Substitutions for DPI Services (continued)

Parameter Name	Description
dpiRules	<p>Defines application rules as a list of map expressions that the policy template can use to bind variables in the policy template to values in this list. See the <b>&lt;for-each-rule&gt;</b> element in the configuration file.</p> <p>The keys in the map expression must be valid identifiers for substitutions. For example, four or fewer hexadecimal digits are interpreted as parts of IPv6 addresses and cannot be used as identifiers.</p> <p>If you do not supply a value, the default value is an empty list.</p>
dpiInterfaceClasses	<p>Specifies the policy templates that should be applied to the interface. See the <b>&lt;target&gt;</b> element in the configuration file.</p> <p>This parameter lets you group your interfaces into classes so that you can specify for each interface which targets should be applied when the DPI script service is triggered. For example, you might have some interfaces with services provided entirely by an MX Series router and some interfaces that have a customer premises router. In your DPI script service configuration, you can have one target to configure the MX Series router with Junos XML management protocol commands and another target to configure the CPE router with Telnet commands.</p> <p>The interface is normally specified as a list of strings in the subscriber hierarchy to define the interfaces affected by the service activation. If you do not supply a value, the default value is an empty list.</p>

**Related  
Documentation**

- [Creating the DPI Script Service \(SRC CLI\) on page 144](#)
- [Configuring Subscriptions to the DPI Script Service on page 145](#)
- [Substituting Parameters in Policy Templates on page 150](#)
- [Configuring Policy Templates on page 151](#)

## Creating a Configuration File

The `/SDK/scriptServices/dpiScriptService/resource/dpiConf.xml` file found in the `SDK+AppSupport+Demos+Samples.tar.gz` file contains a sample configuration file for the script service that demonstrates service activation and deactivation using Junos XML management protocol and Telnet commands.

The configuration file is in the form of an XML document with these sections:

- Optimization parameters—Optional section to optimize batch parameters for committing configuration changes
- Policy templates—Mandatory section for specifying the policy rules that will be added to or removed from network devices during service activation or deactivation

The configuration file has this basic structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dpi-configuration SYSTEM "dpi-configuration.dtd" >
<configuration>
  <batch <!-- batch parameters specified as attributes --> />
  <policy-templates>
    <!-- policy template specification -->
  </policy-templates>
</configuration>
```

Tasks to create a configuration file are:

- [Configuring Batch Parameters on page 150](#)
- [Substituting Parameters in Policy Templates on page 150](#)
- [Configuring Policy Templates on page 151](#)

## Configuring Batch Parameters

To avoid the overhead of individual commits for policy changes that occur close together, the script service groups the policy changes for a network device into a batch so that the commits can happen at the same time.

To optimize the batch parameters, you can specify these timing attributes for the **<batch/>** element:

- **wait-time**—Time to wait for the next command for each device before committing the configuration. The default value is 60 seconds.
- **max-commit-delay**—Maximum time to wait before committing the configuration. The default value is 120 seconds.

For example, to specify a **wait-time** of 15 seconds and **max-commit-delay** of 30 seconds:

```
<batch wait-time="15" max-commit-delay="30" />
```

- See Also**
- [Substituting Parameters in Policy Templates on page 150](#)
  - [Configuring Policy Templates on page 151](#)

## Substituting Parameters in Policy Templates

The SRC software can substitute values for variables in the policy templates. The following variables are supported in the policy templates:

- Attributes of the ServiceSessionInfo object in the SAE script service API
- Variables from parameter acquisition
- Variables defined in the dpiRules parameter that are acquired in the context of a **<for-each-rule>** element





**NOTE:** If the same variable is defined in both the `ServiceSessionInfo` interface and parameter acquisition, the value in the `ServiceSessionInfo` interface is used. However, the value defined in the `dpiRules` parameter override the other values.

For information about the `ServiceSessionInfo` interface, see the script service documentation in the SAE core API documentation on the Juniper Networks website at <https://www.juniper.net/documentation/software/management/src/api-index.html>.

The value of the variable can be used in the policy templates as defined. You can also specify how to use the value of a variable by extracting part of the value from the variable or replacing nonalphanumeric characters in the value with underscores.

- To extract part of the value from the variable, follow the variable with a tilde (~) and a Java regular expression pattern. The regular expression is matched against the value of the variable, and the value of the last capture group is the result of instantiating the variable expression. For more information about using regular expressions, see <http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html>.

For example: `[[[ variable~[^\.]+\.(d+) ]]]`

If we replace *variable* with *interfaceName* and the value of *interfaceName* is *ge-1/2/3.4*, then this expression would evaluate to *4*.

- To replace all nonalphanumeric characters in the value with underscores, follow the variable with an underscore (\_).

For example: `[[[ variable_ ]]]`

If we replace *variable* with *interfaceName* and the value of *interfaceName* is *ge-1/2/3.4*, then this expression would evaluate to *ge\_1\_2\_3\_4*.



**NOTE:** You can use the underscore and the tilde expressions together, but the underscore must precede the tilde in the expression.

**See Also** • [Configuring Policy Templates on page 151](#)

## Configuring Policy Templates

The policy templates are used to define the policy rules that are inserted or removed from network devices. Templates are combined with parameters from the service activation context to generate Junos XML management protocol and Telnet commands that add and remove service policies.

The policy templates section has this basic structure:

```
<policy-templates>
  <target interface-class="<!-- interface class name -->">
```

```

<activation>
  <junoscript>
    <!-- JUNOScript API statements -->
    <for-each-rule>
      <!-- Can have multiple for-each-rule -->
      <if test="expression">
        <!-- Can have conditional expressions -->
      </if>
    </for-each-rule>
    <for-each-rule test="expression">
      <!-- Can have multiple for-each-rule -->
      <!-- For each single rule, can include test conditions -->
    </for-each-rule>
  </junoscript>
  <telnet host="<!-- hostname -->">
    <prompt>login:</prompt>
    <command>joe</command>
    <prompt>password:</prompt>
    <command>abc123</command>
    <!-- Can have many prompt/command pairs -->
    <for-each-rule test="expression">
      <!-- For each single rule, can include conditions
            and have prompt/command pairs -->
    </for-each-rule>
  </telnet>
</activation>
<deactivation>
  <!-- Structure same as for activation -->
</deactivation>
</target>
</policy-templates>

```

Table 11 on page 152 describes the policy template elements in the configuration file.

**Table 11: Policy Template Elements for Configuration File**

Element	Description
<b>&lt;target interface-class="interface-class-name"&gt;</b>	<p>Defines a single policy template, which is selected by matching the <b>interface-class</b> attribute with the value found in the <code>dpiInterfaceClasses</code> parameter. If the <b>interface-class</b> attribute is not provided or its value is "", the target applies to all interfaces.</p> <p>For example: <b>&lt;target interface-class="MXEnterprise"&gt;</b></p>
<b>&lt;activation&gt;</b>	<p>Defines what the script service should do when activating or modifying a session. This element is triggered when the <code>dpiAdminState</code> parameter changes from <b>"disabled"</b> to <b>"enabled"</b>.</p>
<b>&lt;deactivation&gt;</b>	<p>Defines what the script service should do when deactivating a session. This element is triggered when the <code>dpiAdminState</code> parameter changes from <b>"enabled"</b> to <b>"disabled"</b>.</p>
<b>&lt;junoscript&gt;</b>	<p>Contains a sequence of Junos XML management protocol commands to manage policies on routers running Junos OS.</p> <p>This element can contain <b>&lt;if&gt;</b> and <b>&lt;for-each-rule&gt;</b> elements, delimited variables, literal text, and XML elements, which are not interpreted.</p>

Table 11: Policy Template Elements for Configuration File (continued)

Element	Description
<code>&lt;telnet host="hostname"&gt;</code>	<p>Contains a sequence of prompt and command pairs to match on the Telnet device, similar to an expect script. The <b>host</b> attribute is a variable that can include a regular expression to extract a part of the value from the variable. See the <b>&lt;variable&gt;</b> element.</p> <p>For example: <code>&lt;telnet host="deviceIP"&gt;</code></p> <p>This element can contain <b>&lt;if&gt;</b>, <b>&lt;for-each-rule&gt;</b>, <b>&lt;prompt&gt;</b>, and <b>&lt;command&gt;</b> elements. The <b>&lt;prompt&gt;</b> and <b>&lt;command&gt;</b> elements must alternate, and the sequence must start with the <b>&lt;prompt&gt;</b> element. This element can also contain delimited variables and literal text.</p>
<code>&lt;variable-delimiters start="delimiter" end="delimiter"&gt;</code>	<p>Specifies the delimiters for variables in the configuration file. The default delimiters enclose the variable within three square brackets (<code>[[[ variable ]]]</code>).</p> <p>If you want to specify a different delimiter, you must specify the <b>&lt;variable-delimiters&gt;</b> element immediately after the opening tag for the <b>&lt;junoscript&gt;</b> or <b>&lt;telnet&gt;</b> element. The delimiters apply to the contents of the <b>&lt;junoscript&gt;</b> or <b>&lt;telnet&gt;</b> element. Any other occurrences of the <b>&lt;variable-delimiters&gt;</b> element within that element are ignored.</p> <p>For example: <code>&lt;variable-delimiters start="(*" end="*)"&gt;</code></p>
<code>&lt;if test= "variableName~pattern"&gt;</code>	<p>Defines conditional expressions used to generate configuration commands.</p> <p>The <b>test</b> attribute is a variable expression without delimiters. The test is true if the variable has a value and if the optional regular expression matches the variable.</p> <p>For example, the forwarding-class statement would be added to the body only if the map expression contained the fcl key to satisfy the test condition:</p> <pre>&lt;if test="fcl"&gt; forwarding-class [[[ fcl ]]]; &lt;/if&gt;</pre>
<code>&lt;for-each-rule&gt;</code>	<p>Creates the specified body in the policy template for instantiating each map expression found in the <b>dpiRules</b> parameter. For example, if you have two map expressions in the <b>dpiRules</b> parameter, the policy template would generate the body of the <b>&lt;for-each-rule&gt;</b> element once for each map expression.</p> <p>The <b>&lt;for-each-rule&gt;</b> element has a <b>ruleNumber</b> variable to sequentially track the processing of each map expression.</p> <p>You can use the <b>test</b> attribute to provide a condition for the rule; using this attribute would be the same as adding an <b>&lt;if&gt;</b> element.</p>



**NOTE:** When using special XML characters as part of the policy templates, they must be coded in XML. For example, the left angle bracket (<) must be coded as &lt;.

The following example uses some elements to show a policy template that activates application-aware access list (ACL) services and service sets on an MX Series router by loading the configuration in text format using Junos XML management protocol.

```
<policy-templates>
  <target interface-class="MXEnterprise">
    <activation>
      <junoscript>
        <rpc>
          <load-configuration action="replace" format="text">
            <configuration-text>
services {
  acl {
    rule ACL_{{{ interfaceName_ }}} {
      match-direction input-output;
      <for-each-rule>
        term {{{ ruleNumber }}} {
          from {
            application junos:{{{ app }}};
          }
          then {
            <if test="fcl">
              forwarding-class {{{ fcl }}};
            </if>
            <if test="action~accept">
              count application;
            </if>
            {{{ action }}};
          }
        }
      </for-each-rule>
    }
  }
  service-set SSET_{{{ interfaceName_ }}} {
    acl-rules ACL_{{{ interfaceName_ }}};
    interface-service {
      service-interface ms-1/{{{ interfaceName~[^\.]+\-\d+/(^\d+/\d+\.^\d+)
}}];
    }
  }
}
interfaces {
  {{{ interfaceName~[^\.]+\}\.^\d+ }}} {
    unit {{{ interfaceName~[^\.]+\}\.^\d+ }}} {
      family inet {
        service {
          input {
            service-set SSET_{{{ interfaceName_ }}}
          }
          output {
            service-set SSET_{{{ interfaceName_ }}}
          }
        }
      }
    }
  }
}
```

```

    }
  }
}

</configuration-text>
</load-configuration>
</rpc>
</junoscript>
</activation>
</target>
</policy-templates>

```

If the example uses the following dpiRules substitution:

```

dpiRules=[{app="rtsp", action="accept", fcl="expedited-forwarding"},
           {app="bittorrent", action="discard"}]

```

The two map expressions in the dpiRules parameter might generate the following target configuration (with two terms) from the policy template example:

```

services {
  aacl {
    rule AAACL_xe_8_3_0_1001 {
      match-direction input-output;
      term 1 {
        from {
          applications junos:rtsp;
        }
        then {
          forwarding-class expedited-forwarding;
          count application;
          accept;
        }
      }
      term 2 {
        from {
          applications junos:bittorrent;
        }
        then {
          discard;
        }
      }
    }
  }
  service-set SSET_xe_8_3_0_1001 {
    aacl-rules AAACL_([[ interfaceName_ ]]);
    interface-service {
      service-interface ms-1/3/0.1001;
    }
  }
}

interfaces {
  xe-8/3/0 {
    unit 1001 {
      family inet {

```

```
service {
  input {
    service-set SSET_xe_8_3_0_1001;
  }
  output {
    service-set SSET_xe_8_3_0_1001;
  }
}
}
```

- See Also**
- [Substituting Parameters in Policy Templates on page 150](#)
  - For information about the Junos XML management protocol, see *Junos XML Management Protocol Developer Guide*
  - For information about service set configuration on routers running Junos OS, see *Junos OS Services Interfaces Library for Routing Devices*
  - For information about accessing and configuring the third-party device using Telnet, see the device's software documentation

- Related Documentation**
- See *Junos XML Management Protocol Developer Guide*
  - See *Junos OS Services Interfaces Library for Routing Devices*
  - [Creating the DPI Script Service \(SRC CLI\) on page 144](#)

---

## Example: Using the DPI Script Service

To use the DPI script service provided:

1. Download the DPI script service to your system from the Juniper Networks website:  
<https://www.juniper.net/support/downloads/?p=src#sw>

The files for supporting the DPI script service can be found in the SRC Demo and Sample Application software (*SDK+AppSupport+Demos+Samples.tar.gz* file).

The */SDK/scriptServices/dpiScriptService/lib/dpiss.jar* file contains the DPI script service implementation. Copy the *dpiss.jar* file to a location that is accessible to the SAE by a URL.

The */SDK/scriptServices/dpiScriptService/resource/dpiConfig.xml* file contains the sample configuration file that is included in the *dpiss.jar* file.

2. Import the sample data for the DPI script service using an LDAP browser.

The */SDK/scriptServices/dpiScriptService/ldif/dpiService.ldif* and */SDK/scriptServices/dpiScriptService/ldif/dpiSubscriber.ldif* files contain the sample service definition and subscriber configuration for setting up the script service.

To load the sample data into the database, you can use an LDAP tool, such as **ldapmodify**. 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*.

3. Modify the service substitutions for your device.

You can make these substitutions by defining the parameter substitutions in the DPI service with the SRC CLI or by passing the values through the enterprise portal.

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

**Related  
Documentation**

- For information about defining parameter substitutions with the SRC CLI, see [Creating the DPI Script Service \(SRC CLI\) on page 144](#) or [Configuring Subscriptions to the DPI Script Service on page 145](#)
- For information about passing the values through the SAE core API, see [Substituting Parameters in Policy Templates on page 150](#)
- *Subscriptions Overview*





## CHAPTER 15

# Configuring PCC or ePCC Rules for Router Running Junos OS and Acting as PCEF

- [Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF on page 159](#)
- [Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 160](#)
- [Configuring Policies for Router Running Junos OS and Acting as PCEF \(SRC CLI\) on page 162](#)
- [Configuring Policy Lists for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 163](#)
- [Configuring Static PCC Rules for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 165](#)
- [Configuring Substitutions for Gx Static PCC Rules on page 168](#)
- [Configuring Dynamic PCC Rules for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 168](#)
- [Configuring Substitutions for Gx Dynamic PCC Rules on page 172](#)
- [Configuring the Dynamic PCC Rules Application Information for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 174](#)
- [Configuring the Dynamic PCC Rules Flow Information for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 175](#)
- [Configuring the Dynamic PCC Rules QoS Information for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 177](#)
- [Configuring the Dynamic PCC Rules Steering Information for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 178](#)
- [Configuring the Dynamic PCC Rules Redirect Information for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 180](#)

## **Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF**

---

The SRC software acting as PCRF uses the Gx router driver to establish a southbound Gx interface between the SRC software and the MX Series router (that is, Services Control Gateway) acting as PCEF.

The Gx router driver has the following responsibilities:

- Manage subscriber sessions signaled by the Services Control Gateway.
- Activate or 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.
- Notify the SAE when subscribers log out.

The Gx router driver responds to requests from the Services Control Gateway, 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.

The SRC software provisions static PCC rules, dynamic PCC rules, and dynamic ePCC rules to the Services Control Gateway through the Gx router driver using the PULL or PUSH procedure based on subscriber profile configuration. The PCC and ePCC rules provide the policy control and applicable charging information for a service data flow.

In PUSH procedure (unsolicited provisioning of the rules), the SRC software provisions the rules in the RAR message to the Services Control Gateway without receiving any request from the Services Control Gateway.

In PULL procedure (solicited provisioning of the rules), the SRC software provisions the rules in the CCA message to the Services Control Gateway on receiving a request from the Services Control Gateway.

**Related  
Documentation**

- *SAE Support for Gx Router Driver*
- *Managing MX Series Routers Acting as a PCEF Using the SRC Software Overview*
- [Configuring Policies for Router Running Junos OS and Acting as PCEF \(SRC CLI\) on page 162](#)
- [Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 160](#)

---

## Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF (SRC CLI)

---

Use the following configuration statements to configure policies used for routers (Services Control Gateways) running Junos OS and acting as PCEF:

```
policies group name {  
    description description;  
}  
policies group name list name {  
    role junos-gx;  
    applicability both;  
    description description;
```

```

}
policies group name list name rule name {
  type type;
  precedence precedence;
  accounting;
}
policies group name list name rule name static-pcc-rule {
  charging-rule-name charging-rule-name;
  charging-rule-base-name charging-rule-base-name;
  description description;
}
policies group name list name rule name dynamic-pcc-rule {
  charging-rule-name charging-rule-name;
  mute-notification;
  flow-status (ENABLED-UPLINK | ENABLED-DOWNLINK | ENABLED | DISABLED |
    REMOVED);
  forwarding-class-name forwarding-class-name;
  LRF-profile-name LRF-profile-name;
  HCM-profile-name HCM-profile-name;
  online;
  reporting-level (SERVICE-IDENTIFIER-LEVEL | RATING-GROUP-LEVEL |
    SPONSORED-CONNECTIVITY-LEVEL);
  description description;
}
policies group name list name rule name dynamic-pcc-rule application-information {
  TDF-application-id TDF-application-id;
  TDF-application-id-base TDF-application-id-base;
}
policies group name list name rule name dynamic-pcc-rule gx-flows name {
  flow-description flow-description;
  tos-traffic-class tos-traffic-class;
  security-parameter-index security-parameter-index;
  flow-label flow-label;
  flow-direction (UNSPECIFIED | DOWNLINK | UPLINK | BIDIRECTIONAL);
}
policies group name list name rule name dynamic-pcc-rule qos-information {
  max-requested-bw-UL max-requested-bw-UL;
  max-requested-bw-DL max-requested-bw-DL;
}
policies group name list name rule name dynamic-pcc-rule steering-information {
  service-chain-identifier service-chain-identifier;
  steering-uplink-VRF steering-uplink-VRF;
  steering-downlink-VRF steering-downlink-VRF;
  steering-ip-address steering-ip-address;
  keep-existing-steering (STEERING-ENABLED | STEERING-DISABLED);
}
policies group name list name rule name dynamic-pcc-rule redirect-information {
  redirect-address-type (IPv4-Address | IPv6-Address | URL | SIP-URL);
  redirect-server-address redirect-server-address;
}

```

**Related  
Documentation**

- [Configuring Policies for Router Running Junos OS and Acting as PCEF \(SRC CLI\) on page 162](#)

- [Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF on page 159](#)

## Configuring Policies for Router Running Junos OS and Acting as PCEF (SRC CLI)

---

The role of the policy list for the Services Control Gateway must be set to **junos-gx**. The policy list must be configured to contain the rule of type **gx-static-pcc-rule** or **gx-dynamic-pcc-rule**.

Before you configure policies for the Services Control Gateway, 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 policies for Services Control Gateway:

1. Create a policy group.

For information about creating the policy group, see *Configuring Policy Groups (SRC CLI)*.

2. Configure the policy list and set the role of the list to **junos-gx** and the **applicability** option to **both**.

For information about configuring the policy list, see “[Configuring Policy Lists for Routers Running Junos OS and Acting as PCEF \(SRC CLI\)](#)” on page 163.

3. Configure the policy rule and set the rule type to **gx-static-pcc-rule** or **gx-dynamic-pcc-rule**.

For information about configuring the policy rule, see “[Configuring Static PCC Rules for Routers Running Junos OS and Acting as PCEF \(SRC CLI\)](#)” on page 165 and “[Configuring Dynamic PCC Rules for Routers Running Junos OS and Acting as PCEF \(SRC CLI\)](#)” on page 168.

4. Configure the additional details for the dynamic PCC rule or dynamic ePCC rule.

For information about configuring additional details for the dynamic rules, see the following topics:

- [Configuring the Dynamic PCC Rules Application Information for Routers Running Junos OS and Acting as PCEF \(SRC CLI\)](#) on page 174
- [Configuring the Dynamic PCC Rules Flow Information for Routers Running Junos OS and Acting as PCEF \(SRC CLI\)](#) on page 175

- [Configuring the Dynamic PCC Rules QoS Information for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 177](#)
- [Configuring the Dynamic PCC Rules Steering Information for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 178](#)
- [Configuring the Dynamic PCC Rules Redirect Information for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 180](#)



**NOTE:** If the ADC parameters (mute notification, redirect information, and TDF information) are configured for the dynamic PCC rule, then the rule is called as dynamic ePCC rule.

#### Related Documentation

- [Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 160](#)
- [Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF on page 159](#)

## Configuring Policy Lists for Routers Running Junos OS and Acting as PCEF (SRC CLI)

Use the following configuration statements to configure policy lists for routers (Services Control Gateways) running Junos OS and acting as PCEF.



**NOTE:** To configure policy lists for router (Services Control Gateway) acting as PCEF, you must:

- Set the role of the policy list to **junos-gx**
- Set the policy list rule type to **gx-static-pcc-rule** or **gx-dynamic-pcc-rule**
- Set the policy list **applicability** option to **both**

```
policies group name list name {
  role junos-gx;
  applicability both;
  description description;
}
```

To configure policy lists:

1. From configuration mode, create a policy list. For example, to create a policy list called **gx-list** within a policy group called **GXnew**:

```
user@host# edit policies group GXnew list gx-list
```

2. Set the role of the policy list to **junos-gx**.

```
[edit policies group GXnew list gx-list]
user@host# set role junos-gx
```

3. Specify where the policy is applied on the device. The **applicability** option must be set to **both**.

```
[edit policies group GXnew list gx-list]
user@host# set applicability both
```

4. (Optional) Specify the description for the policy list.

```
[edit policies group GXnew list gx-list]
user@host# set description description
```

5. (Optional) Modify the policy substitutions for your Gx policies.

Gx policy attributes allow the value substitutions with parameters.

For information about configuring the substitutions for Gx static and dynamic PCC rules, see [“Configuring Substitutions for Gx Static PCC Rules” on page 168](#) and [“Configuring Substitutions for Gx Dynamic PCC Rules” on page 172](#).

6. (Optional) Verify your configuration.

```
[edit policies group GXnew list]
user@host# show
gx-list {
  applicability both;
  role junos-gx;
  rule dynpcc-rule-name {
    dynamic-pcc-rule {
      LRF-profile-name LRF-profile-name;
      application-information {
        TDF-application-id TDF-application-id;
        TDF-application-id-base TDF-application-id-base;
      }
    }
    charging-rule-name Testnew;
    flow-status ENABLED;
    forwarding-class-name forwarding-class-name;
    gx-flows {
      flow1 {
        flow-description flow-description;
        flow-direction BIDIRECTIONAL;
        flow-label flow-label;
        security-parameter-index security-parameter-index;
        tos-traffic-class tos-traffic-class;
      }
      flow2 {
        flow-description flow-description;
        flow-direction UPLINK;
        flow-label flow-label;
        security-parameter-index security-parameter-index;
        tos-traffic-class tos-traffic-class;
      }
    }
  }
}
```

```

    }
  }
  mute-notification;
  online;
  qos-information {
    max-requested-bw-DL max-requested-bw-DL;
    max-requested-bw-UL max-requested-bw-UL;
  }
  redirect-information {
    redirect-address-type IPv4-Address;
    redirect-server-address redirect-server-address;
  }
  reporting-level RATING-GROUP-LEVEL;
  steering-information {
    keep-existing-steering STEERING-ENABLED;
    service-chain-identifier service-chain-identifier;
    steering-downlink-VRF steering-downlink-VRF;
    steering-ip-address steering-ip-address;
    steering-uplink-VRF steering-uplink-VRF;
  }
}
precedence precedence;
type gx-dynamic-pcc-rule;
}
rule statpcc-rule-name {
  accounting;
  static-pcc-rule {
    charging-rule-name cname;
  }
  type gx-static-pcc-rule;
}
}

```

#### Related Documentation

- [Configuring Policies for Router Running Junos OS and Acting as PCEF \(SRC CLI\) on page 162](#)
- [Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 160](#)
- [Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF on page 159](#)

## Configuring Static PCC Rules for Routers Running Junos OS and Acting as PCEF (SRC CLI)

Use the following configuration statements to configure static PCC rules, which enable the SRC software to provision policies (activated only using name) to the routers (Services Control Gateways) acting as PCEF and running Junos OS through the Gx interface by using the Gx router driver.



**NOTE:** You can create multiple static PCC rules. You can configure the usage monitoring information for the service having static PCC rules even though the SRC software does not support monitoring key association for the static PCC rules.

For creating a static PCC rule, you must:

- Set the role of the policy list to **junos-gx**
- Set the policy list rule type to **gx-static-pcc-rule**
- Set the policy list applicability option to **both**

```
.....  
policies group name list name rule name {  
    type type;  
    precedence precedence;  
    accounting;  
}  
policies group name list name rule name static-pcc-rule {  
    charging-rule-name charging-rule-name;  
    charging-rule-base-name charging-rule-base-name;  
    description description;  
})  
.....
```



**NOTE:** Precedence mapping is not supported for the static PCC rule.

- .....
1. From configuration mode, create a static PCC rule inside a policy list that has already been created and configured. For example, to create a static PCC rule called `statpcc-rul1-name` within a policy list called `gx-list`:

```
[edit]  
user@host# edit policies group GXnew list gx-list rule statpcc-rul1-name
```

2. Set the type of policy rule to **gx-static-pcc-rule**.

```
[edit policies group GXnew list gx-list rule statpcc-rul1-name]  
user@host# set type gx-static-pcc-rule
```

3. (Optional) Enable the accounting flag so that the SRC software requests the usage monitoring information from the Services Control Gateway.





**NOTE:** If you enable the accounting functionality for a rule in the policy group, the accounting functionality is enabled for all rules in the policy group.

When you enable the accounting functionality, you must configure the `USAGE_REPORT` event trigger for the subscriber profiles and configure the granted service unit for the subscribed services.

```
[edit policies group GXnew list gx-list rule statpcc-rul1-name]
user@host# set accounting
```

4. From configuration mode, access the configuration statement that configures the static PCC rule.

```
[edit]
user@host# edit policies group GXnew list gx-list rule statpcc-rul1-name static-pcc-rule
```

5. (Optional) Specify a static PCC rule name.



**NOTE:** You must configure either rule name or rule base name for the static PCC rule. The rule name should be unique for each IP CAN session.

```
[edit policies group GXnew list gx-list rule statpcc-rul1-name static-pcc-rule]
user@host# set charging-rule-name charging-rule-name
```

6. (Optional) Specify a name of a PCC rule group residing at the Services Control Gateway.



**NOTE:** You must configure either rule name or rule base name for the static PCC rule. The rule base name should be unique for each IP CAN session.

```
[edit policies group GXnew list gx-list rule statpcc-rul1-name static-pcc-rule]
user@host# set charging-rule-base-name charging-rule-base-name
```

7. (Optional) Specify a description for the static PCC rule.

```
[edit policies group GXnew list gx-list rule statpcc-rul1-name static-pcc-rule]
user@host# set description description
```

8. (Optional) Verify your configuration.

```
[edit policies group GXnew list gx-list rule statpcc-rule-name]
user@host# show
accounting;
precedence precedence;
static-pcc-rule {
    charging-rule-name crname;
}
type gx-static-pcc-rule;
```

#### Related Documentation

- [Configuring Policies for Router Running Junos OS and Acting as PCEF \(SRC CLI\) on page 162](#)
- [Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 160](#)
- [Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF on page 159](#)

## Configuring Substitutions for Gx Static PCC Rules

Table 12 on page 168 lists the parameters qualified for configuring substitutions for Gx static PCC rules.

**Table 12: Substitutions for Gx Static PCC Rules**

Parameter Name	Description
charging-rule-name	Defines a name for the charging rule.
charging-rule-base-name	Defines a name for a predefined group of static PCC rules residing at the PCEF.

#### Related Documentation

- [Configuring Policies for Router Running Junos OS and Acting as PCEF \(SRC CLI\) on page 162](#)
- [Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 160](#)
- [Configuring Static PCC Rules for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 165](#)
- [Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF on page 159](#)

## Configuring Dynamic PCC Rules for Routers Running Junos OS and Acting as PCEF (SRC CLI)

Use the following configuration statements to configure dynamic PCC rules, which enable the SRC software to provision dynamic policies to the routers (Services Control Gateways)

acting as PCEF and running Junos OS through the Gx interface by using the Gx router driver.



**NOTE:** You can create multiple dynamic PCC rules. For creating a dynamic PCC rule, you must:

- Set the role of the policy list to `junos-gx`
- Set the policy list rule type to `gx-dynamic-pcc-rule`
- Set the policy list applicability option to `both`



**NOTE:** If the mute notification, application information, and redirect information are configured, then the rule is called as ePCC rule.

```

policies group name list name rule name {
  type type;
  precedence precedence;
  accounting;
}
policies group name list name rule name dynamic-pcc-rule {
  charging-rule-name charging-rule-name;
  mute-notification;
  flow-status (ENABLED-UPLINK | ENABLED-DOWNLINK | ENABLED | DISABLED |
    REMOVED);
  forwarding-class-name forwarding-class-name;
  LRF-profile-name LRF-profile-name;
  HCM-profile-name HCM-profile-name;
  online;
  reporting-level (SERVICE-IDENTIFIER-LEVEL | RATING-GROUP-LEVEL |
    SPONSORED-CONNECTIVITY-LEVEL);
  description description;
}

```

1. From configuration mode, create a dynamic PCC rule inside a policy list that has already been created and configured. For example, to create a dynamic PCC rule called `dynpcc-rul1-name` within a policy list called `gx-list`:

```

[edit]
user@host# edit policies group GXnew list gx-list rule dynpcc-rul1-name

```

2. Set the type of policy rule to `gx-dynamic-pcc-rule`.

```

[edit policies group GXnew list gx-list rule dynpcc-rul1-name]
user@host# set type gx-dynamic-pcc-rule

```

3. (Optional) Specify the order in which the service data flow templates are applied when service data flow is detected at the Services Control Gateway. The value ranges from 1 through 65,535.



**NOTE:** The precedence value should be unique for each IP CAN session.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name]
user@host# set precedence precedence
```

4. (Optional) Enable the accounting flag so that the SRC software requests the usage monitoring information from the Services Control Gateway.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name]
user@host# set accounting
```

5. From configuration mode, access the configuration statement that configures the dynamic PCC rule.

```
[edit]
user@host# edit policies group GXnew list gx-list rule dynpcc-rul1-name
dynamic-pcc-rule
```

6. Specify a dynamic PCC rule name.



**NOTE:** The rule name should be unique for each IP CAN session.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule]
user@host# set charging-rule-name charging-rule-name
```

7. (Optional) Disable sending the PCEF application start or stop notification to the SRC software.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule]
user@host# set mute-notification
```

By default, the PCEF application start or stop notification is sent to the SRC software.

8. (Optional) Specify the traffic flow status.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule]
user@host# set flow-status (ENABLED-UPLINK | ENABLED-DOWNLINK | ENABLED
| DISABLED | REMOVED)
```

9. (Optional) Specify the name of the forwarding class. This value is transmitted between the Services Control Gateway and SRC software through the Juniper Networks VSA (Forwarding-Class-Name).

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule]
user@host# set forwarding-class-name forwarding-class-name
```

10. (Optional) Specify the name of the LRF profile. This value is transmitted between the Services Control Gateway and SRC software through the Juniper Networks VSA (LRF-Profile-Name).

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule]
user@host# set LRF-profile-name LRF-profile-name
```

11. (Optional) Specify the name of the HCM profile. This value is transmitted between the Services Control Gateway and SRC software through the Juniper Networks VSA (HCM-Profile-Name).

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule]
user@host# set HCM-profile-name HCM-profile-name
```

12. (Optional) Specify whether the online charging interface provided by the Services Control Gateway can be used for the dynamic PCC rule.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule]
user@host# set online
```

By default, the online charging interface configured at the Services Control Gateway is used.

13. (Optional) Specify a level at which the Services Control Gateway should report the usage information.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule]
user@host# set reporting-level (SERVICE-IDENTIFIER-LEVEL | RATING-GROUP-LEVEL
| SPONSORED-CONNECTIVITY-LEVEL)
```

14. (Optional) Specify a description for the dynamic PCC rule.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule]
user@host# set description description
```

15. (Optional) Verify your configuration.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule]
user@host# show
LRF-profile-name LRF-profile-name;
```

```

application-information {
    TDF-application-id TDF-application-id;
    TDF-application-id-base TDF-application-id-base;
}
charging-rule-name Testnew;
flow-status ENABLED;
forwarding-class-name forwarding-class-name;
gx-flows {
    flow1 {
        flow-description flow-description;
        flow-direction BIDIRECTIONAL;
        flow-label flow-label;
        security-parameter-index security-parameter-index;
        tos-traffic-class tos-traffic-class;
    }
    flow2 {
        flow-description flow-description;
        flow-direction UPLINK;
        flow-label flow-label;
        security-parameter-index security-parameter-index;
        tos-traffic-class tos-traffic-class;
    }
}
mute-notification;
online;
qos-information {
    max-requested-bw-DL max-requested-bw-DL;
    max-requested-bw-UL max-requested-bw-UL;
}
redirect-information {
    redirect-address-type IPv4-Address;
    redirect-server-address redirect-server-address;
}
reporting-level RATING-GROUP-LEVEL;
steering-information {
    keep-existing-steering STEERING-ENABLED;
    service-chain-identifier service-chain-identifier;
    steering-downlink-VRF steering-downlink-VRF;
    steering-ip-address steering-ip-address;
    steering-uplink-VRF steering-uplink-VRF;
}

```

#### Related Documentation

- [Configuring Policies for Router Running Junos OS and Acting as PCEF \(SRC CLI\) on page 162](#)
- [Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 160](#)
- [Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF on page 159](#)

## Configuring Substitutions for Gx Dynamic PCC Rules

Table 13 on page 173 lists the parameters qualified for configuring substitutions for Gx dynamic PCC rules.

**Table 13: Substitutions for Gx Dynamic PCC Rules**

Parameter Name	Description
HCM-profile-name	Specifies the name of the HCM profile on the Services Control Gateway.
LRF-profile-name	Specifies the name of the LRF profile on the Services Control Gateway.
charging-rule-name	Defines a name for the dynamic PCC rule.
forwarding-class-name	Specifies the name of the forwarding class on the Services Control Gateway.
flow-description	Defines a packet filter for an IP flow.
flow-label	Specifies an IPv6 flow label header.
security-parameter-index	Specifies the security parameter index of an IPSec packet.
tos-traffic-class	Defines the IPv4 ToS and ToS mask or IPv6 traffic class and traffic class mask.
max-requested-bw-DL	Specifies a maximum bit rate for downlink.
max-requested-bw-UL	Specifies a maximum bit rate for uplink.
redirect-server-address	Specifies the address of the redirect server with which the end user should be connected when the account cannot cover the service cost on the Services Control Gateway.
service-chain-identifier	Identifies the service chain on the Services Control Gateway.
steering-downlink-VRF	Specifies the VRF information about the steering downlink on the Services Control Gateway.
steering-ip-address	Specifies the IP address of the steering interface on the Services Control Gateway.
steering-uplink-VRF	Specifies the VRF information about the steering uplink on the Services Control Gateway.

**Related Documentation**

- [Configuring Dynamic PCC Rules for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 168](#)
- [Configuring Policies for Router Running Junos OS and Acting as PCEF \(SRC CLI\) on page 162](#)
- [Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 160](#)

- [Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF on page 159](#)

## Configuring the Dynamic PCC Rules Application Information for Routers Running Junos OS and Acting as PCEF (SRC CLI)

---

Use the following configuration statements to configure the application information for the dynamic PCC rules. For more information about creating a dynamic PCC rule, see “[Configuring Dynamic PCC Rules for Routers Running Junos OS and Acting as PCEF \(SRC CLI\)](#)” on page 168.

```
policies group name list name rule name dynamic-pcc-rule application-information {  
    TDF-application-id TDF-application-id;  
    TDF-application-id-base TDF-application-id-base;  
}
```

1. From configuration mode, access the configuration statements that configure application information for the dynamic PCC rules. This procedure uses GXnew as the policy group, gx-list as the policy list, and dynpcc-rul1-name as the dynamic PCC rule.

```
[edit]  
user@host# edit policies group GXnew list gx-list rule dynpcc-rul1-name  
dynamic-pcc-rule application-information
```

2. (Optional) Specify the ID of the PCEF application for which the ADC rules are applied.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule  
application-information]  
user@host# set TDF-application-id TDF-application-id
```



**NOTE:** You must specify the application ID that is supported by the Services Control Gateway.

3. (Optional) Specify the name for a group of PCEF applications.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule  
application-information]  
user@host# set TDF-application-id-base TDF-application-id-base
```



**NOTE:** You must specify the application base name that is supported by the Services Control Gateway.

4. (Optional) Verify your configuration.



```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule
application-information]
user@host# show
TDF-application-id TDF-application-id;
TDF-application-id-base TDF-application-id-base;
```

#### Related Documentation

- [Configuring Policies for Router Running Junos OS and Acting as PCEF \(SRC CLI\) on page 162](#)
- [Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 160](#)
- [Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF on page 159](#)

## Configuring the Dynamic PCC Rules Flow Information for Routers Running Junos OS and Acting as PCEF (SRC CLI)

Use the following configuration statements to configure the flow information for the dynamic PCC rules. For more information about creating a dynamic PCC rule, see “Configuring Dynamic PCC Rules for Routers Running Junos OS and Acting as PCEF (SRC CLI)” on page 168.

```
policies group name list name rule name dynamic-pcc-rule gx-flows name {
  flow-description flow-description;
  tos-traffic-class tos-traffic-class;
  security-parameter security-parameter;
  flow-label flow-label;
  flow-direction flow-direction;
}
```

1. From configuration mode, access the configuration statements that configure flow information for the dynamic PCC rules. This procedure uses GXnew as the policy group, gx-list as the policy list, dynpcc-rul1-name as the dynamic PCC rule, and flow1 as the flow information name.

```
[edit]
user@host# edit policies group GXnew list gx-list rule dynpcc-rul1-name
dynamic-pcc-rule gx-flows flow1
```

2. (Optional) Specify a packet filter for an IP flow.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule gx-flows
flow1]
user@host# set flow-description flow-description
```

3. (Optional) Specify the IPv4 ToS and ToS mask or the IPv6 traffic class and traffic class mask.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule gx-flows
flow1]
user@host# set tos-traffic-class tos-traffic-class
```

4. (Optional) Specify the security parameter index of a packet.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule gx-flows
flow1]
user@host# set security-parameter-index security-parameter-index
```

5. (Optional) Specify an IPv6 flow label header.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule gx-flows
flow1]
user@host# set flow-label flow-label
```

6. (Optional) Specify a direction for which the filter is applicable.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule gx-flows
flow1]
user@host# set flow-direction (UNSPECIFIED | DOWNLINK | UPLINK | BIDIRECTIONAL)
```

7. (Optional) Verify your configuration.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule
gx-flows]
user@host# show
flow1 {
    flow-description flow-description;
    flow-direction BIDIRECTIONAL;
    flow-label flow-label;
    security-parameter-index security-parameter-index;
    tos-traffic-class tos-traffic-class;
}
flow2 {
    flow-description flow-description;
    flow-direction UPLINK;
    flow-label flow-label;
    security-parameter-index security-parameter-index;
    tos-traffic-class tos-traffic-class;
}
```

#### Related Documentation

- [Configuring Policies for Router Running Junos OS and Acting as PCEF \(SRC CLI\) on page 162](#)
- [Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 160](#)
- [Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF on page 159](#)

## Configuring the Dynamic PCC Rules QoS Information for Routers Running Junos OS and Acting as PCEF (SRC CLI)

Use the following configuration statements to configure the QoS information for the dynamic PCC rules. For more information about creating a dynamic PCC rule, see “Configuring Dynamic PCC Rules for Routers Running Junos OS and Acting as PCEF (SRC CLI)” on page 168.

```
policies group name list name rule name dynamic-pcc-rule qos-information {
    max-requested-bw-UL max-requested-bw-UL;
    max-requested-bw-DL max-requested-bw-DL;
}
```

1. From configuration mode, access the configuration statements that configure QoS information for the dynamic PCC rules. This procedure uses GXnew as the policy group, gx-list as the policy list, and dynpcc-rul1-name as the dynamic PCC rule.

```
[edit]
user@host# edit policies group GXnew list gx-list rule dynpcc-rul1-name
dynamic-pcc-rule qos-information
```

2. (Optional) Specify a maximum bit rate for uplink. The value ranges from 1 through 256,000.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule
qos-information]
user@host# set max-requested-bw-UL max-requested-bw-UL
```

3. (Optional) Specify a maximum bit rate for downlink. The value ranges from 1 through 256,000.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule
qos-information]
user@host# set max-requested-bw-DL max-requested-bw-DL
```

4. (Optional) Verify your configuration.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule
qos-information]
user@host# show
max-requested-bw-DL max-requested-bw-DL;
max-requested-bw-UL max-requested-bw-UL;
```

### Related Documentation

- [Configuring Policies for Router Running Junos OS and Acting as PCEF \(SRC CLI\) on page 162](#)

- [Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF \(SRC CLI\)](#) on page 160
- [Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF](#) on page 159

## Configuring the Dynamic PCC Rules Steering Information for Routers Running Junos OS and Acting as PCEF (SRC CLI)

---

Use the following configuration statements to configure the steering information for the dynamic PCC rules. These steering information are transmitted between the SRC software and Services Control Gateway through the Juniper Networks VSAs. For more information about creating a dynamic PCC rule, see [“Configuring Dynamic PCC Rules for Routers Running Junos OS and Acting as PCEF \(SRC CLI\)”](#) on page 168.

```
policies group name list name rule name dynamic-pcc-rule steering-information {  
  service-chain-identifier service-chain-identifier;  
  steering-uplink-VRF steering-uplink-VRF;  
  steering-downlink-VRF steering-downlink-VRF;  
  steering-ip-address steering-ip-address;  
  keep-existing-steering (STEERING-ENABLED | STEERING-DISABLED);  
}
```

1. From configuration mode, access the configuration statements that configure PCEF steering information for the dynamic PCC rules. This procedure uses GXnew as the policy group, gx-list as the policy list, and dynpcc-rul1-name as the dynamic PCC rule.

```
[edit]  
user@host# edit policies group GXnew list gx-list rule dynpcc-rul1-name  
dynamic-pcc-rule steering-information
```

2. (Optional) Specify the service chain identifier. This value is transmitted between the Services Control Gateway and SRC software through the Juniper Networks VSA (Service-Chain-Identifier).

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule  
steering-information]  
user@host# set service-chain-identifier service-chain-identifier
```

3. (Optional) Specify the VRF information about the steering uplink. This value is transmitted between the Services Control Gateway and SRC software through the Juniper Networks VSA (Steering-Uplink-VRF).

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule  
steering-information]  
user@host# set steering-uplink-VRF steering-uplink-VRF
```

4. (Optional) Specify the VRF information about the steering downlink. This value is transmitted between the Services Control Gateway and SRC software through the Juniper Networks VSA (Steering-Downlink-VRF).

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule
steering-information]
user@host# set steering-downlink-VRF steering-downlink-VRF
```

5. (Optional) Specify the IP address. This value is transmitted between the Services Control Gateway and SRC software through the Juniper Networks VSA (Steering-IP-Address).

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule
steering-information]
user@host# set steering-ip-address steering-ip-address
```

6. (Optional) Specify whether to keep the existing steering information or not. This value is transmitted between the Services Control Gateway and SRC software through the Juniper Networks VSA (Keep-Existing-Steering).

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule
steering-information]
user@host# set keep-existing-steering (STEERING-ENABLED | STEERING-DISABLED)
keep-existing-steering
```

7. (Optional) Verify your configuration.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule
steering-information]
user@host# show
keep-existing-steering STEERING-ENABLED;
service-chain-identifier service-chain-identifier;
steering-downlink-VRF steering-downlink-VRF;
steering-ip-address steering-ip-address;
steering-uplink-VRF steering-uplink-VRF;
```

#### Related Documentation

- [Configuring Policies for Router Running Junos OS and Acting as PCEF \(SRC CLI\) on page 162](#)
- [Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 160](#)
- [Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF on page 159](#)

## Configuring the Dynamic PCC Rules Redirect Information for Routers Running Junos OS and Acting as PCEF (SRC CLI)

Use the following configuration statements to configure the redirect information for the dynamic PCC rules. For more information about creating a dynamic PCC rule, see “Configuring Dynamic PCC Rules for Routers Running Junos OS and Acting as PCEF (SRC CLI)” on page 168.

```
policies group name list name rule name dynamic-pcc-rule redirect-information {
  redirect-address-type (IPv4-Address | IPv6-Address | URL | SIP-URL);
  redirect-server-address redirect-server-address;
}
```

1. From configuration mode, access the configuration statements that configure redirect information for the dynamic PCC rules. This procedure uses GXnew as the policy group, gx-list as the policy list, and dynpcc-rul1-name as the dynamic PCC rule.

```
[edit]
user@host# edit policies group GXnew list gx-list rule dynpcc-rul1-name
dynamic-pcc-rule redirect-information
```

2. (Optional) Specify the address type.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule
redirect-information]
user@host# set redirect-address-type (IPv4-Address | IPv6-Address | URL | SIP-URL)
```

3. (Optional) Specify the address of the redirect server with which the end user should be connected when the account cannot cover the service cost.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule
redirect-information]
user@host# set redirect-server-address redirect-server-address
```

4. (Optional) Verify your configuration.

```
[edit policies group GXnew list gx-list rule dynpcc-rul1-name dynamic-pcc-rule
redirect-information]
user@host# show
redirect-address-type IPv4-Address;
redirect-server-address redirect-server-address;
```

### Related Documentation

- [Configuring Policies for Router Running Junos OS and Acting as PCEF \(SRC CLI\) on page 162](#)
- [Configuration Statements for Policies Used for Routers Running Junos OS and Acting as PCEF \(SRC CLI\) on page 160](#)

- [Managing PCC or ePCC Rules on Routers Running Junos OS and Acting as PCEF on page 159](#)





## PART 4

# Using SRC Configuration Wizards

- [SRC Configuration Wizards Overview \(SRC CLI\) on page 185](#)
- [SRC Configuration Wizards Overview \(C-Web Interface\) on page 189](#)
- [Using the Fair Usage on MX Series Routers Configuration Wizard on page 193](#)



# SRC Configuration Wizards Overview (SRC CLI)

- [SRC Configuration Wizards Overview \(SRC CLI\) on page 185](#)
- [Running a Configuration Wizard \(SRC CLI\) on page 187](#)

## SRC Configuration Wizards Overview (SRC CLI)

---

The SRC software includes configuration *wizards* to simplify configuring the most common configuration scenarios. Each configuration wizard uses an XML definition file that generates a specific configuration scenario. Most of the configuration is predefined in the definition file. However, because each configuration scenario is unique, definition files cannot predefine all options, so the wizard prompts you for input specific to your implementation.

### How Configuration Wizards Work (SRC CLI)

You can invoke a configuration wizard from the SRC CLI. At runtime, the configuration wizard processes the definition file and presents the corresponding configuration steps. The interface prompts you to enter information for any options specific to your configuration that are not predefined in the definition file. The values you enter are used for the respective parameters in the definition file. After you enter all required parameters, the interface displays a list of SRC CLI set commands corresponding to the parameters you entered. After you review the configuration, you can either select Commit to commit the configuration or you can select Back to make changes to the parameters.

While running a configuration wizard, if you close the wizard midway, the uncommitted configurations are saved to a temporary file called the *tag* file. The naming convention of a tag file is **<wizard definition filename>\_<username>\_CLItag\_<timestamp>.tmp**.

Where:

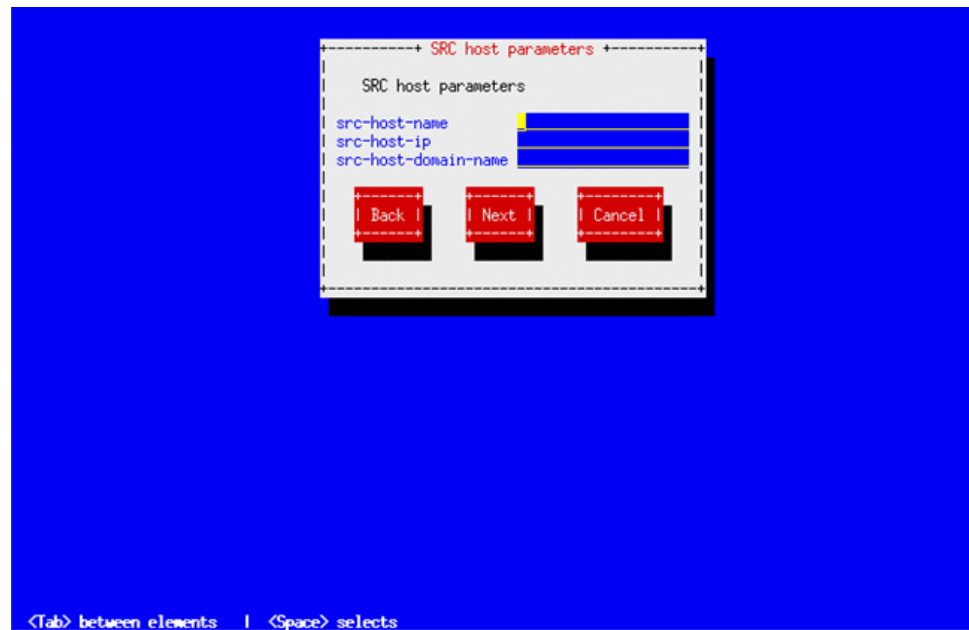
- *wizard definition filename*—Specifies the name of the wizard definition file.
- *username*—Specifies the name of the user.
- *timestamp*—Specifies the current system timestamp.

You can resume the configuration at any time later by using the **[configuration-wizard wizard-name tag tag-file-name]** command. When you commit the configuration changes, the saved tag file is automatically deleted.

## Navigating Screens in the Wizard (SRC CLI)

The wizard interface consists of buttons, which you navigate using the keyboard. [Figure 15 on page 186](#) shows a sample screen for the configuration wizard. Because each wizard configures a different scenario, each wizard is unique.

*Figure 15: Sample SRC Configuration Wizard Screen (SRC CLI)*



[Table 14 on page 186](#) and [Table 15 on page 187](#) list the buttons and navigation keys for the configuration wizard.

*Table 14: Wizard Buttons (SRC CLI)*

Button	Description
Back	Go to the previous step.
Next	Go to the next step.
Cancel	Stop the execution of the command <b>[configuration-wizard wizard-name tag tag-file-name]</b> .
Finish	Select this button only after you configure all arguments.
Commit	Commit the wizard configuration.

Table 15: Wizard Navigation Keys

Key	Description
Tab	Move between buttons (elements) such as Back, Next, Cancel, Commit, Finish, and so on.
Space bar	Select the highlighted button.

- Related Documentation**
- [Running a Configuration Wizard \(SRC CLI\) on page 187](#)
  - [SRC Configuration Wizards Overview \(C-Web Interface\) on page 189](#)

## Running a Configuration Wizard (SRC CLI)

To run a configuration wizard:

- From configuration mode, access the configuration statement that runs the configuration wizard.

```
user@host# configuration-wizard wizard-name wizard-name tag tag-file-name
```

Where:

- *wizard-name*—Specifies the name of the wizard you want to run.
- *tag-file-name*—Specifies the name of a tag file that is automatically generated when you save the uncommitted configurations.

- Related Documentation**
- [SRC Configuration Wizards Overview \(SRC CLI\) on page 185](#)
  - [Running a Configuration Wizard \(C-Web Interface\) on page 191](#)



# SRC Configuration Wizards Overview (C-Web Interface)

- [SRC Configuration Wizards Overview \(C-Web Interface\)](#) on page 189
- [Running a Configuration Wizard \(C-Web Interface\)](#) on page 191

## SRC Configuration Wizards Overview (C-Web Interface)

---

The C-Web configuration wizard enables you to enter the most common configuration scenarios and prompts you for input specific to your configuration scenario. At the end of the wizard, you can either commit or discard the configuration changes that are displayed in a tree-like format.

### How the Configuration Wizards Work (C-Web Interface)

You can invoke a configuration wizard from the SRC C-Web interface. The configuration wizard uses the standard or customized wizard definition file as input to group and present related configurations on a single page of the wizard. You can upload the customized definition file by navigating to the specific wizard definition file from your environment. You can also edit the wizard definition file, upload the modified file, and customize the configuration wizard display by using the C-Web interface. In the pop-up wizard, each configuration step is presented in the order in which it is defined in the wizard definition file. A step is composed of closely knit configuration inputs for the same component. A collection defines repetitive steps in the wizard, which enables you to use multiple instances of the configuration setup. Collections helps you easily configure multiple instances.

When you close the configuration wizard pop-up, a confirmation pop-up asks you if you want to save the uncommitted changes. On confirmation, the uncommitted changes are saved in a temporary file called the *tag* file. The naming convention of a tag file is **<wizard definition filename>\_<username>\_Cwebtag\_<timestamp>.tmp**.

Where:

- *wizard definition filename*—Specifies the name of the wizard definition file.
- *username*—Specifies the name of the user.
- *timestamp*—Specifies the current system timestamp.

To resume configuration at a later time, you need to specify the tag filename and the wizard name. When you commit the configuration changes, the saved tag file is automatically deleted.

## Navigating Screens in the Wizard (C-Web Interface)

You can navigate around the wizard pop-up using the **next** and **back** buttons. On each wizard pop-up, enter the configuration data. When you finish configuring the data on all wizard pop-ups, click **Finish** to view the modified configuration changes along with any validation errors. The commit button is unavailable if any validation error is displayed in the wizard. In this case, click << **Go Back** to navigate to the wizard pop-up for making changes in the configuration. When no validation errors are displayed in the modified configuration wizard, you can click **Commit** after checking the configuration to commit your changes. [Figure 16 on page 190](#) shows a sample screen for the C-Web configuration wizard.

*Figure 16: Sample SRC Configuration Wizard Screen (C-Web Interface)*

[Table 16 on page 190](#) lists the buttons for the configuration wizard pop-up.

*Table 16: Wizard Pop-up Buttons*

Button	Description
back	Go to the previous step.
next	Go to the next step.
Close	Close the wizard pop-up.



Table 16: Wizard Pop-up Buttons (continued)

Button	Description
Create New Instance	Create a new collection instance.  <b>NOTE:</b> You can view this button only if the configuration inputs are part of a collection.
Commit	Commit the wizard configuration.  <b>NOTE:</b> This button is displayed only in the final wizard pop-up containing the modified configuration tree.
<<Go Back	Navigate back to the configuration wizard pop-up.  <b>NOTE:</b> This button is displayed only in the final wizard pop-up containing the modified configuration tree.
Finish	Display the modified configuration wizard in a tree-like format.  <b>NOTE:</b> This button is displayed on the last page of the wizard pop-up.

- Related Documentation**
- [Running a Configuration Wizard \(C-Web Interface\) on page 191](#)
  - [SRC Configuration Wizards Overview \(SRC CLI\) on page 185](#)

## Running a Configuration Wizard (C-Web Interface)

The configuration wizard simplifies configuring the most common configuration scenarios.

To run a configuration wizard using the C-Web Interface:

1. Click **Configure > Configuration Wizard**.

The configuration wizard appears.



**NOTE:**

- You can access the configuration wizard only if your privilege is set as **superuser**.
- You are prompted to commit or roll back the configuration, if there are any previous uncommitted configuration changes.

2. Select an option button to locate the standard or custom wizard definition file.
  - To open the standard definition file, enter the name of the wizard definition file you want to run.
  - To upload the customized definition file, click the **Browse** button and navigate to the specific wizard definition file from your environment. Then click **Upload**.



**NOTE:** Incorrect changes to the wizard definition file can result in undesired configuration changes. Any customization of the wizard definition file must be approved by Juniper Networks.

3. In the **Tag File** text box, enter the name of the tag file.



**NOTE:**

- The naming convention of a tag file is `<wizard definition filename>_<username>_Cwebtag_<timestamp>.tmp`.

Where:

- *wizard definition filename*—Specifies the name of the wizard definition file.
- *username*—Specifies the name of the user.
- *timestamp*—Specifies the current system timestamp.
- You can access the configuration wizard along with the tag file when the tag file is compatible with the wizard definition file. These files are compatible only if the tag file is created from the corresponding wizard definition file.

4. Click **Open wizard**. The configuration wizard is displayed. On each wizard pop-up, enter the configuration data.
5. Click **Finish** on the final wizard pop-up. The modified CLI data tree is displayed.
6. Click **Commit** to commit your changes.

You can navigate back to the wizard and correct any validation errors by using the **<<Go Back** button.

**Related  
Documentation**

- [SRC Configuration Wizards Overview \(C-Web Interface\) on page 189](#)
- [Running a Configuration Wizard \(SRC CLI\) on page 187](#)

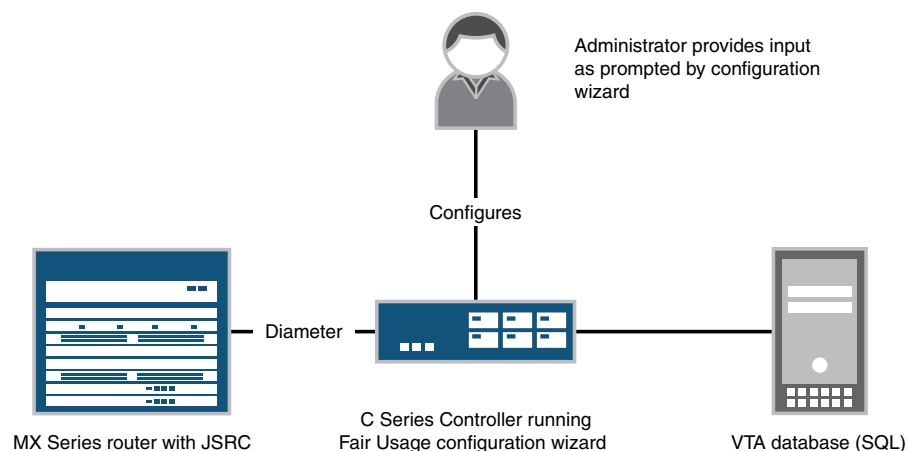
# Using the Fair Usage on MX Series Routers Configuration Wizard

- Fair Usage on MX Series Routers Configuration Wizard Overview on page 193
- Fair Usage on MX Series Routers Configuration Wizard Configuration Overview on page 194
- Running the Fair Usage on MX Series Routers Configuration Wizard (SRC CLI) on page 205

## Fair Usage on MX Series Routers Configuration Wizard Overview

The fair usage on MX Series routers configuration wizard creates the SRC configuration shown in [Figure 17 on page 193](#). In the default configuration created by the wizard, each SRC VTA subscriber is initialized with a certain amount of periodic quota but no purchased quota. The SAE maps all subscribers to a single subscriber profile, which has both high-speed and low-speed service subscriptions. The high-speed service, called MXQuotaInternet, operates at 10 Mbps and is activated when the subscriber logs in. The MXQuotaInternet service continues to run until the subscriber's quota is exhausted. When the quota is exhausted, the subscriber is switched to the low-speed service called MXQuotaLowSpeed, which operates at 256 Kbps.

*Figure 17: Fair Usage on MX Series Routers Configuration Wizard Topology*



g041261

The fair usage on MX Series routers configuration wizard requires one C Series Controller, one SQL database, and one MX Series router. The wizard configures a single SRC host. The wizard does not configure the SQL database or the MX Series router; you must configure these separately in order for them to work with the configuration created by the wizard.

The SRC policies and service substitution configured by the wizard refer to certain Junos OS dynamic profile names and firewall filter names. You must configure these names on the MX Series router for the configuration to work properly. Alternatively, you can modify these names in the SRC CLI to match those configured on the MX Series router after you run the wizard and commit the SRC configuration.

The wizard configures the SRC VTA component to use a MySQL database. This database must be deployed on a separate host and you must create the database by using the "vta-database-mysql.sql" file, which is included with the SRC VTA component. The wizard requires you to enter the database host IP address, database username, and password.

The wizard also requires you to enter the SRC hostname, IP address, and domain name, as well as the router hostname, IP address, and domain name. These parameters are essential for the configuration of Diameter peers and SAE-managed devices.

**Related  
Documentation**

- [Fair Usage on MX Series Routers Configuration Wizard Configuration Overview on page 194](#)
- [Running the Fair Usage on MX Series Routers Configuration Wizard \(SRC CLI\) on page 205](#)

---

## Fair Usage on MX Series Routers Configuration Wizard Configuration Overview

When you use the fair usage on MX Series routers configuration wizard, the wizard definition file specifies most of the SRC configuration. However, you must provide values for certain parameters in the configuration.

### Fair Usage on MX Series Routers Configuration Wizard Definition File

The fair usage on MX Series routers configuration wizard definition file is an .xml file that controls the parameters of the wizard.

The following sample shows the definition file for the fair usage on MX Series routers configuration wizard. The first part of the definition file defines the dialog boxes you use to enter values specific to your environment. The next part of the definition file lists the SRC CLI set commands the wizard invokes.

This file is shown only for reference purposes. Modification of definition files is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cli SYSTEM "configuration-wizard.dtd">
<cli name="fair-usage-mx">
  <help>Configure SRC for fair usage MX scenario</help>
```

```

<step name="src-config">
  <step name="src-host-parameters">
    <caption>SRC host parameters</caption>
    <help>SRC host parameters</help>
    <description>
      This step collect SRC host parameters that will be used
      in configuration. The parameters are SRC host names, IP addresses
      and domain names.
    </description>
    <wiz-argument name="src-host-name" mandatory="true">
      <help>SRC host name</help>
      <description>SRC host name.</description>
    </wiz-argument>
    <wiz-argument name="src-host-ip" mandatory="true" type="inet">
      <help>SRC host IP address</help>
      <description>SRC host IP address used to communicate with diameter
      peers and VTA database.</description>
    </wiz-argument>
    <wiz-argument name="src-host-domain-name" mandatory="true">
      <help>SRC host domain name</help>
      <description>SRC host domain name used as origin-realm for
      diameter.</description>
    </wiz-argument>
  </step>
  <step name="vta-db-host-parameters">
    <caption>VTA database host and database parameters</caption>
    <help>VTA database host and database parameters</help>
    <description>
      This step collect VTA database host and database parameters that
      will be used
      in configuration. The parameters are VTA database host IP addresses,
      database
      connection user name and password.
    </description>
    <wiz-argument name="vta-database-ip" mandatory="true" type="inet">
      <help>VTA database IP address</help>
      <description>The IP address of the host where VTA database
      runs.</description>
    </wiz-argument>
    <wiz-argument name="vta-database-user" mandatory="true">
      <help>VTA database connection user name</help>
      <description>The user name for VTA to connect to VTA
      database.</description>
    </wiz-argument>
    <wiz-argument name="vta-database-passwd" mandatory="true" type="passwd">
      <help>VTA database connection password</help>
      <description>The password for VTA to connect to VTA
      database.</description>
    </wiz-argument>
  </step>
  <step name="router-host-parameters">
    <caption>Router host parameters</caption>
    <help>Router host parameters</help>
    <description>
      This step collect router host parameters that will be used
      in configuration. The parameters are router host names, IP addresses
      and domain names.
    </description>
  </step>

```

```

    </description>
    <wiz-argument name="mx-router-name" mandatory="true">
      <help>Router host name</help>
      <description>Router host name used as diameter peer origin-host
and SRC network device name.</description>
    </wiz-argument>
    <wiz-argument name="mx-router-ip" mandatory="true" type="inet">
      <help>Router IP address</help>
      <description>Router IP address used for diameter and SRC network
device address.</description>
    </wiz-argument>
    <wiz-argument name="mx-router-domain-name" mandatory="true">
      <help>Router domain name</help>
      <description>Router domain name used as diameter peer
origin-realm.</description>
    </wiz-argument>
    <configuration>
      set system diameter active-peers
      set system diameter local-address {src-config src-host-parameters
src-host-ip}
      set system diameter origin-host {src-config src-host-parameters
src-host-name}
      set system diameter origin-realm {src-config src-host-parameters
src-host-domain-name}
      set system diameter port 3868
      set system diameter protocol tcp

      set shared network diameter peer {src-config router-host-parameters
mx-router-name} active-peer
      set shared network diameter peer {src-config router-host-parameters
mx-router-name} address {src-config router-host-parameters mx-router-ip}
      set shared network diameter peer {src-config router-host-parameters
mx-router-name} connect-timeout 10
      set shared network diameter peer {src-config router-host-parameters
mx-router-name} origin-host {src-config router-host-parameters mx-router-name}

      set shared network diameter peer {src-config router-host-parameters
mx-router-name} port 3868
      set shared network diameter peer {src-config router-host-parameters
mx-router-name} protocol tcp

      set shared network device {src-config router-host-parameters
mx-router-name} description 'A MX fair usage device'
      set shared network device {src-config router-host-parameters
mx-router-name} device-type junos-ise
      set shared network device {src-config router-host-parameters
mx-router-name} management-address {src-config router-host-parameters
mx-router-ip}
      set shared network device {src-config router-host-parameters
mx-router-name} origin-host {src-config router-host-parameters mx-router-name}

      set shared network device {src-config router-host-parameters
mx-router-name} peers {src-config router-host-parameters mx-router-name}
      set shared network device {src-config router-host-parameters
mx-router-name} virtual-router * sae-connection {src-config src-host-parameters
src-host-ip}

      set policies folder fair-usage-ise group MXCaptivePolicy list
captive-list applicability both
      set policies folder fair-usage-ise group MXCaptivePolicy list

```

```

captive-list role junos-ise
    set policies folder fair-usage-ise group MXCaptivePolicy list
captive-list rule rule-1 type junos-ise
    set policies folder fair-usage-ise group MXCaptivePolicy list
captive-list rule rule-1 dynamic-profile profile-name src_driven_captive_profile

    set policies folder fair-usage-ise group MXQuotaPolicy description
'Quota Policy'
    set policies folder fair-usage-ise group MXQuotaPolicy list quota_list
applicability both
    set policies folder fair-usage-ise group MXQuotaPolicy list quota_list
role junos-ise
    set policies folder fair-usage-ise group MXQuotaPolicy list quota_list
rule rule-1 accounting
    set policies folder fair-usage-ise group MXQuotaPolicy list quota_list
rule rule-1 type junos-ise
    set policies folder fair-usage-ise group MXQuotaPolicy list quota_list
rule rule-1 dynamic-profile profile-name src_driven_quota_profile
    set policies folder fair-usage-ise group MXQuotaPolicy list quota_list
rule rule-1 dynamic-profile variables input type any
    set policies folder fair-usage-ise group MXQuotaPolicy list quota_list
rule rule-1 dynamic-profile variables input value ingress_filter_to_use
    set policies folder fair-usage-ise group MXQuotaPolicy list quota_list
rule rule-1 dynamic-profile variables output type any
    set policies folder fair-usage-ise group MXQuotaPolicy list quota_list
rule rule-1 dynamic-profile variables output value egress_filter_to_use
    set policies folder fair-usage-ise group MXQuotaPolicy local-parameters
egress_filter_to_use type any
    set policies folder fair-usage-ise group MXQuotaPolicy local-parameters
ingress_filter_to_use type any

    set services global service MXQuotaInternet accounting-interim-interval
600
    set services global service MXQuotaInternet available
    set services global service MXQuotaInternet category Internet
    set services global service MXQuotaInternet description 'MX quota high
speed service, supposed to be used as VTA behaving service'
    set services global service MXQuotaInternet policy-group
/fair-usage-ise/MXQuotaPolicy
    set services global service MXQuotaInternet radius-class MXQuotaInternet

    set services global service MXQuotaInternet status active
    set services global service MXQuotaInternet tracking-plug-in quotavta

    set services global service MXQuotaInternet type normal
    set services global service MXQuotaLowSpeed accounting-interim-interval
600
    set services global service MXQuotaLowSpeed available
    set services global service MXQuotaLowSpeed category Internet
    set services global service MXQuotaLowSpeed description 'MX quota low
speed service, supposed to be used as VTA misbehaving service'
    set services global service MXQuotaLowSpeed policy-group
/fair-usage-ise/MXQuotaPolicy
    set services global service MXQuotaLowSpeed radius-class MXQuotaInternet

    set services global service MXQuotaLowSpeed status active
    set services global service MXQuotaLowSpeed tracking-plug-in quotavta

    set services global service MXQuotaLowSpeed type normal
    set services global service MXCaptive available

```

```
set services global service MXCaptive description 'MX captive service,
supposed to be used as VTA misbehaving service'
set services global service MXCaptive policy-group
/fair-usage-ise/MXCaptivePolicy
set services global service MXCaptive radius-class MXCaptive
set services global service MXCaptive status active
set services global service MXCaptive tracking-plugin quotavta
set services global service MXCaptive type normal

set subscribers retailer fair-usage-mx domain-name fair-usage-mx.com
set subscribers retailer fair-usage-mx subscriber-folder local
subscriber quota-subscriber-1 common-name One
set subscribers retailer fair-usage-mx subscriber-folder local
subscriber quota-subscriber-1 surname Quotasubscriber
set subscribers retailer fair-usage-mx subscriber-folder local
subscriber quota-subscriber-1 subscription MXCaptive activation manual
set subscribers retailer fair-usage-mx subscriber-folder local
subscriber quota-subscriber-1 subscription MXCaptive status active
set subscribers retailer fair-usage-mx subscriber-folder local
subscriber quota-subscriber-1 subscription MXQuotaInternet activation
automatically-on-login
set subscribers retailer fair-usage-mx subscriber-folder local
subscriber quota-subscriber-1 subscription MXQuotaInternet status active
set subscribers retailer fair-usage-mx subscriber-folder local
subscriber quota-subscriber-1 subscription MXQuotaInternet substitution [
'egress_filter_to_use=\"10m-service\"' 'ingress_filter_to_use=\"10m-service\"'
]
set subscribers retailer fair-usage-mx subscriber-folder local
subscriber quota-subscriber-1 subscription MXQuotaLowSpeed activation manual
set subscribers retailer fair-usage-mx subscriber-folder local
subscriber quota-subscriber-1 subscription MXQuotaLowSpeed status active
set subscribers retailer fair-usage-mx subscriber-folder local
subscriber quota-subscriber-1 subscription MXQuotaLowSpeed substitution [
'egress_filter_to_use=\"256k-service\"' 'ingress_filter_to_use=\"256k-service\"'
]

set slot 0 nic scenario-name OnePopLogin
set slot 0 nic snmp-agent

set slot 0 sae shared /SAE/fair-usage

set shared sae group fair-usage configuration plug-ins name quotavta
ejb-adaptor jndi-sae-event-listener vta-Quota/SAEEventListenerBean
set shared sae group fair-usage configuration plug-ins name quotavta
ejb-adaptor application-server-url {src-config src-host-parameters
src-host-ip}:1099
set shared sae group fair-usage configuration plug-ins name quotavta
ejb-adaptor ejb-clustering-strategy EJBObjectClustering
set shared sae group fair-usage configuration plug-ins name quotavta
ejb-adaptor jndi-service-provider org.jnp.interfaces.NamingContextFactory
set shared sae group fair-usage configuration plug-ins name nic external
corba-object-reference corbaname::{src-config src-host-parameters
src-host-ip}:2809/NameService#nicsae/saePort
set shared sae group fair-usage configuration plug-ins event-publishers
subscriber-tracking [fileAcct quotavta nic ]
set shared sae group fair-usage configuration plug-ins event-publishers
service-tracking [fileAcct quotavta ]
set shared sae group fair-usage subscriber-classifier rule rule-mx
target
uniqueID=quota-subscriber-1,ou=local,retailerName=fair-usage-mx,o=Users,o=umc
```



```

        set shared sae group fair-usage subscriber-classifier rule rule-mx
condition nasPortId==ge-*
        insert shared sae group fair-usage subscriber-classifier rule rule-mx
before rule-1

        set slot 0 application-server web virtual-host eth0 alias [ {src-config
src-host-parameters src-host-name} {src-config src-host-parameters src-host-ip}
]

        set shared vta group Quota nic-proxy IdToSaeNicProxy
        set shared vta group fairusage subscriber-id-solution login-name
        set shared vta group fairusage action CalcUsage function
db-engine-calculate-usage
        set shared vta group fairusage action CalculateInterim function
db-engine-calculate-interim
        set shared vta group fairusage action DebitAccounts function
db-engine-update-accounts
        set shared vta group fairusage action DebitAccounts parameter
script-name DebitQuotaUsage
        set shared vta group fairusage action GetAccountBalances function
db-engine-get-accounts
        set shared vta group fairusage action SetInterim function
sae-set-interim-interval
        set shared vta group fairusage action SetInterim parameter
current-subscriber-only
        set shared vta group fairusage action StartCaptiveService function
sae-start-service
        set shared vta group fairusage action StartCaptiveService parameter
current-subscriber-only
        set shared vta group fairusage action StartCaptiveService parameter
subscription-name MXCaptive
        set shared vta group fairusage action StartQuotaInternetService function
sae-start-service
        set shared vta group fairusage action StartQuotaInternetService
parameter current-subscriber-only
        set shared vta group fairusage action StartQuotaInternetService
parameter subscription-name MXQuotaInternet
        set shared vta group fairusage action StartQuotaLowSpeedService function
sae-start-service
        set shared vta group fairusage action StartQuotaLowSpeedService
parameter current-subscriber-only
        set shared vta group fairusage action StartQuotaLowSpeedService
parameter subscription-name MXQuotaLowSpeed
        set shared vta group fairusage action StopQuotaInternetService function
sae-stop-service
        set shared vta group fairusage action StopQuotaInternetService parameter
current-subscriber-only
        set shared vta group fairusage action StopQuotaInternetService parameter
subscription-name MXQuotaInternet
        set shared vta group fairusage action StopQuotaLowSpeedService function
sae-stop-service
        set shared vta group fairusage action StopQuotaLowSpeedService parameter
current-subscriber-only
        set shared vta group fairusage action StopQuotaLowSpeedService parameter
subscription-name MXQuotaLowSpeed
        set shared vta group fairusage action TerminateSession function
db-engine-terminate-session
        set shared vta group fairusage database check-valid-connection-sql
'select 1'

```

```

        set shared vta group fairusage database connection-url
jdbc:mysql://{src-config vta-db-host-parameters vta-database-ip}:3306/quotavta

        set shared vta group fairusage database datasource-mapping mySQL
        set shared vta group fairusage database driver-class
com.mysql.jdbc.Driver
        set shared vta group fairusage database max-pool-size 50
        set shared vta group fairusage database min-pool-size 5
        set shared vta group fairusage database password {src-config
vta-db-host-parameters vta-database-passwd}
        set shared vta group fairusage database user-name {src-config
vta-db-host-parameters vta-database-user}
        set shared vta group fairusage event-handler GetQuota actions
GetAccountBalances
        set shared vta group fairusage event-handler GetQuota events [
service-start:MXQuotaInternet service-start:MXQuotaLowSpeed
service-interim:MXQuotaInternet service-interim:MXQuotaLowSpeed
service-stop:MXQuotaInternet service-stop:MXQuotaLowSpeed account-update ]
        set shared vta group fairusage event-handler GetQuota priority 1
        set shared vta group fairusage event-handler RecordUsage actions [
CalcUsage DebitAccounts ]
        set shared vta group fairusage event-handler RecordUsage events [
service-interim:MXQuotaInternet service-interim:MXQuotaLowSpeed
service-stop:MXQuotaInternet service-stop:MXQuotaLowSpeed ]
        set shared vta group fairusage event-handler RecordUsage priority 5
        set shared vta group fairusage event-handler SetInterim actions [
CalculateInterim SetInterim ]
        set shared vta group fairusage event-handler SetInterim condition
'return <balance_PeriodicQuota>+<balance_BoughtQuota>&0;'
        set shared vta group fairusage event-handler SetInterim events [
service-start:MXQuotaInternet service-start:MXQuotaLowSpeed
service-interim:MXQuotaInternet service-interim:MXQuotaLowSpeed ]
        set shared vta group fairusage event-handler SetInterim priority 10
        set shared vta group fairusage event-handler NoQuota actions [
StopQuotaInternetService StartQuotaLowSpeedService ]
        set shared vta group fairusage event-handler NoQuota condition 'return
<balance_PeriodicQuota>+<balance_BoughtQuota>&0;'
        set shared vta group fairusage event-handler NoQuota events [
service-start:MXQuotaInternet service-interim:MXQuotaInternet ]
        set shared vta group fairusage event-handler NoQuota priority 15
        set shared vta group fairusage event-handler QuotaRefilled actions [
StopQuotaLowSpeedService StartQuotaInternetService ]
        set shared vta group fairusage event-handler QuotaRefilled condition
'var newBalance=<balance_BoughtQuota>+<balance_PeriodicQuota>;\n
if(<old_balance_PeriodicQuota>==null)
<old_balance_PeriodicQuota>=<balance_PeriodicQuota>;\n
if(<old_balance_BoughtQuota>==null)
<old_balance_BoughtQuota>=<balance_BoughtQuota>;\n return
<old_balance_PeriodicQuota>+<old_balance_BoughtQuota>&0&
&newBalance>0;'
        set shared vta group fairusage event-handler QuotaRefilled events
account-update
        set shared vta group fairusage event-handler QuotaRefilled priority
20
        set shared vta group fairusage event-handler EndofBilling actions
TerminateSession
        set shared vta group fairusage event-handler EndofBilling events
callback:TERMINATESESSION
        set shared vta group fairusage event-handler EndofBilling priority 25

```

```

set shared vta group fairusage processor db-engine record-balance-change

set shared vta group fairusage processor db-engine account BoughtQuota
initial-balance 0
set shared vta group fairusage processor db-engine account BoughtQuota
initial-status Inactive
set shared vta group fairusage processor db-engine account PeriodicQuota
initial-balance 100000000000
set shared vta group fairusage processor db-engine account PeriodicQuota
initial-status Active
set shared vta group fairusage processor db-engine account-update-script
DebitQuotaUsage script ' var newPeriodicBalance=0;\n var
newBoughtBalance=0;\n if(&lt;currentUsage&gt;=&lt;balance_PeriodicQuota&gt;){\n
newPeriodicBalance=&lt;balance_PeriodicQuota&gt;-&lt;currentUsage&gt;;\n
newBoughtBalance=&lt;balance_BoughtQuota&gt;;\n } \n else
if(&lt;currentUsage&gt;&lt;=&lt;balance_PeriodicQuota&gt;+&lt;balance_BoughtQuota&gt;){\n
newBoughtBalance=&lt;balance_BoughtQuota&gt;-(&lt;currentUsage&gt;
-&lt;balance_PeriodicQuota&gt;);\n newPeriodicBalance=0;\n } \n
if(newPeriodicBalance!=&lt;balance_PeriodicQuota&gt;){\n
&lt;balance_PeriodicQuota&gt;=newPeriodicBalance;\n
&lt;lastUpdateTime_PeriodicQuota&gt;=&lt;currentTime&gt;;\n } \n
if(newBoughtBalance!=&lt;balance_BoughtQuota&gt;){\n
&lt;balance_BoughtQuota&gt;=newBoughtBalance;\n
&lt;lastUpdateTime_BoughtQuota&gt;=&lt;currentTime&gt;;\n }'
set shared vta group fairusage processor db-engine service
MXQuotaInternet interim-interval-function 'return Math.min(7200,Math.max(900,
(&lt;balance_PeriodicQuota&gt;+&lt;balance_BoughtQuota&gt;)/250000));'
set shared vta group fairusage processor db-engine service
MXQuotaInternet usage-metric-function 'return Math.min(7200,Math.max(900,
(&lt;balance_PeriodicQuota&gt;+&lt;balance_BoughtQuota&gt;)/250000));'
set shared vta group fairusage processor db-engine service
MXQuotaLowSpeed interim-interval-function 'return Math.min(7200,Math.max(900,
(&lt;balance_PeriodicQuota&gt;+&lt;balance_BoughtQuota&gt;)/250000));'
set shared vta group fairusage processor db-engine service
MXQuotaLowSpeed usage-metric-function 'return Math.min(7200,Math.max(900,
(&lt;balance_PeriodicQuota&gt;+&lt;balance_BoughtQuota&gt;)/250000));'
set shared vta group fairusage queue max-concurrency 50
set shared vta group fairusage queue max-queue-size 100000
</configuration>
</step>
</step>
</cli>

```

The following sections describe which parameters are predefined by the wizard and which parameters require your input.

## Configuration Provided by the Fair Usage on MX Series Routers Configuration Wizard

The fair usage on MX Series routers configuration wizard configures the SRC components and configuration trees described in [Table 17 on page 202](#). Most of the SRC configuration is defined by the wizard definition file. Some parameters are configured based on your inputs to the configuration wizard interface.

**Table 17: SRC Configuration Parameters Supplied by the Fair Usage on MX Series Routers Configuration Wizard**

SRC Component or Configuration Tree	Description
Web application server (appsvr)	The wizard configures the Web application server, web virtual host eth0 alias, based on the input you provide for the src-host-name and src-host-ip parameters.
Diameter application	<p>The wizard configures the Diameter application based on the values you enter for the following SRC host parameters in the wizard interface:</p> <ul style="list-style-type: none"> <li>Origin-host—The wizard configures the Diameter origin-host based on the value you enter for the src-host-name parameter.</li> <li>Origin-realm—The wizard configures the Diameter origin-realm based on the value you enter for the src-host-domain-name parameter.</li> <li>Local-address—The wizard configures the Diameter local-address based on the value you enter for the src-host-ip parameter.</li> </ul>
Diameter peer	<p>The wizard configures the Diameter peer parameters for the MX Series router based on the values you enter for the following router host parameters:</p> <ul style="list-style-type: none"> <li>Diameter peer address—The wizard configures the IP address for the Diameter peer based on the value you enter for the mx-router-ip parameter.</li> <li>Diameter peer origin-host—The wizard configures the origin-host of the Diameter peer based on the value you enter for the mx-router-name.</li> </ul> <p>For the connection to the Diameter peer, the wizard uses TCP, port 3868, and specifies the connection as active.</p>
NIC	The wizard configures the NIC to use the OnePopLogin scenario.
SAE	<p>The wizard uses the SAE group name “fair-usage”, which must exist before invoking the wizard. If the group “fair-usage” does not exist, you must create it by committing <b>set slot 0 sae shared /SAE/fair-usage</b>. Because the wizard uses a single-step configuration commit process, it is not possible to commit the local and shared SAE configurations simultaneously. The wizard configures the following parameters for the SAE:</p> <ul style="list-style-type: none"> <li>SRC VTA ejb-adaptor plug-ins</li> <li>NIC plug-ins</li> <li>Plug-ins event publisher</li> <li>SAE logger</li> <li>Subscriber classifier. All subscribers are classified to a single SRC VTA user profile.</li> </ul>

**Table 17: SRC Configuration Parameters Supplied by the Fair Usage on MX Series Routers Configuration Wizard (continued)**

SRC Component or Configuration Tree	Description
VTA	<p><i>Services</i>—The wizard configures an SRC VTA group called “fair-usage” and two services. The high-speed service, called MXQuotaInternet, operates at 10 Mbps and is activated when the subscriber logs in. The MXQuotaInternet service continues to run until the subscriber’s quota is exhausted. When the quota is exhausted, the subscriber is switched to the low-speed service called MXQuotaLowSpeed, which operates at 256 Kbps. The real service behavior depends on the MX Series router firewall filter configuration.</p> <p><i>External MySQL database</i>—The wizard requires an external MySQL database, which you must configure. The wizard configures basic parameters for the external database based on the values you specify in the wizard VTA database host and database parameters dialog box. The database connection-url is based on the value you enter for the vta-database-ip parameter. The username and password are based on the values you enter for vta-database-user and vta-database-pass. If you want to use a database other than MySQL, you must customize the configuration using the SRC CLI after you commit the configuration using the wizard.</p> <p><i>SRC VTA NIC proxy</i>—The wizard uses the IdToSaeNicProxy NIC proxy, which uses the subscriber ID to locate the SAE when NIC resolution is needed. The wizard also configures the NIC to use the OnePopLogin scenario.</p> <p><i>Event Handlers</i>—The wizard configures event handlers so that when there is no quota left for a subscriber, its MXQuotaInternet service is stopped and the MXQuotaLowSpeed service is started. When the subscriber’s quota is refilled, the MXQuotaLowSpeed service is stopped and the MXQuotaInternet service is restarted.</p> <p><i>db-engine processor</i>—The wizard configures the db-engine processor with some initial balance in the PeriodQuota account. The SRC VTA quota account is debited according to the DebitQuotaUsage script.</p> <p><i>Actions</i>—The wizard configures actions to retrieve account balances, debit accounts, calculate usage, and start and stop services.</p>
Policies	<p>The wizard creates a policy folder named “fair-usage-ise” and two policies. The service policy named “MXQuotaPolicy” uses a dynamic profile named “src_driven_quota_profile”. The policy named “MXCaptivePolicy” uses a dynamic profile named “src_driven_captive_profile.”</p>
Services	<p>The wizard configures two services. The high-speed service, called MXQuotaInternet, operates at 10 Mbps and is activated when the subscriber logs in. The MXQuotaInternet service continues to run until the subscriber’s quota is exhausted. When the quota is exhausted, the subscriber is switched to the low-speed service called MXQuotaLowSpeed, which operates at 256 Kbps.</p> <p>Both of the services use quota as the tracking plug-in.</p>
Subscribers	<p>The wizard configures one subscriber named “quota-subscriber-1” that subscribes to both the MXQuotaInternet and the MXQuotaLowSpeed services.</p>

**Table 17: SRC Configuration Parameters Supplied by the Fair Usage on MX Series Routers Configuration Wizard (continued)**

SRC Component or Configuration Tree	Description
shared network device (configuration tree)	<p>The shared network device configuration tree sets up the MX Series router so that it can be managed by the SAE. The wizard specifies the router as a junos-ise type device and adds “A MX fair usage device” as the device description.</p> <p>The management-address assigned by the wizard is based on the value you enter for the mx-router-ip parameter.</p> <p>The origin-host assigned by the wizard is based on the value you enter for the mx-router-name parameter.</p> <p>The wizard specifies the router as a peer to the SAE based on the value you enter for the mx-router-name.</p> <p>The wizard specifies the IP address of the SAE that manages the router based on the value you enter for the src-host-ip parameter.</p>

## Required Input Parameters for the Fair Usage on MX Series Routers Configuration Wizard

The fair usage on MX Series routers configuration wizard requires you to input certain parameters that are specific to your environment. When you run the wizard, you are prompted to enter these parameters. [Table 18 on page 204](#) describes these parameters in detail.

**Table 18: Input Parameters Required by the Fair Usage on MX Series Routers Configuration Wizard**

Type of Parameters	Parameters
SRC host parameters	<ul style="list-style-type: none"> <li>src-host-name—The wizard uses the value you enter as the origin-host for the SRC Diameter application.</li> <li>src-host-ip—The wizard uses the value you enter as the local-address for the SRC Diameter application.</li> <li>src-host-domain-name—The wizard uses the value you enter as the origin-realm for the SRC Diameter application.</li> </ul>
Database host and database parameters used by the SRC VTA	<p>The wizard configures the SRC VTA component to use an external MySQL database, but it does not configure the external database. You must deploy the database on a separate host and create the database by using the “vta-database-mysql.sql” file, which is included with the SRC software. The wizard requires you to enter the following parameters for the external database used by the SRC VTA:</p> <ul style="list-style-type: none"> <li>vta-database-ip—The wizard uses the value you enter as the database connection-url.</li> <li>vta-database-user—The wizard uses the value you enter as the database username.</li> <li>vta-database-pass—The wizard uses the value you enter as the database password.</li> </ul>

**Table 18: Input Parameters Required by the Fair Usage on MX Series Routers Configuration Wizard (continued)**

Type of Parameters	Parameters
Router host parameters	<p>The wizard does not configure the MX Series router. The wizard prompts you to enter values to define the MX Series router as a Diameter peer. The wizard requires you to enter the following parameters for the router host:</p> <ul style="list-style-type: none"> <li>mx-router-name—The wizard uses the value you enter as the Diameter peer origin-hostname for the router.</li> <li>mx-router-ip—The wizard uses the value you enter as the Diameter peer IP address.</li> <li>mx-router-domain-name—The wizard uses the value you enter as the Diameter peer name.</li> </ul> <p>You must manually configure the rest of the configuration for the MX Series router.</p> <p>Certain router configuration parameters, such as dynamic profiles, must be consistent between the SRC configuration and the router configuration. The MX Series router configuration described in this section is for reference only.</p> <p>A sample MX Series router configuration that can work with the SRC configuration created by the fair usage on MX Series routers configuration wizard is provided in a text file in the format of the Junos OS configuration. The sample configuration contains everything under the fair-usage group. The configuration configures IP dynamic interfaces for Dynamic Host Configuration Protocol (DHCP) access. You must edit the interface names that are used for DHCP access in this sample configuration. The sample configuration file is <b>DemosAndSamplesApplications/wizard/fair-usage-mx.txt</b> in <b>SDK+AppSupport+Demos+Samples.tar.gz</b>.</p>
Related Documentation	<ul style="list-style-type: none"> <li><a href="#">Fair Usage on MX Series Routers Configuration Wizard Overview on page 193</a></li> <li><a href="#">Running the Fair Usage on MX Series Routers Configuration Wizard (SRC CLI) on page 205</a></li> </ul>

## Running the Fair Usage on MX Series Routers Configuration Wizard (SRC CLI)

The fair usage on MX Series routers configuration wizard uses the SAE group name “fair-usage”, which must exist before you invoke the wizard. If the group “fair-usage” does not exist, you must create it by committing **set slot 0 sae shared /SAE/fair-usage**. Because the wizard uses a single-step configuration commit process, it is not possible to commit the local and shared SAE configurations simultaneously.

Refer to [“Fair Usage on MX Series Routers Configuration Wizard Configuration Overview” on page 194](#) for more information about the parameters you need to configure for this procedure.

The wizard definition file is located under `/opt/UMC/cli/ddl/`. To run the fair usage on MX Series routers configuration wizard:

- From configuration mode, access the configuration statement that runs the fair usage on MX Series routers configuration wizard.

```
[edit]
user@host# configuration-wizard wizard-name fair-usage-mx.wiz.xml
```

Most of the SRC configuration is predefined in the wizard definition file. However, you must enter the values for parameters specific to your environment in the following steps. Navigate through the dialog boxes by pressing the Tab key and the Space bar.

2. Enter the values for the SRC host parameters dialog box. [Figure 18 on page 206](#) shows a sample dialog box for these parameters.

*Figure 18: SRC Host Parameters Dialog Box*

```
+-----+ SRC host parameters +-----+
|
| This step collect SRC host parameters that
| will be used
|           in configuration. The parameters
| are SRC host names, IP addresses
| and domain names.
|
| src-host-name      my-src
| src-host-ip        10.227.2.101
| src-host-domain-name my-src-domain
|
| [Back] [Next] [Cancel]
|
+-----+
```

3. Press Tab to highlight Next and press the Space bar to navigate to the next screen.
4. Enter the values for the VTA database host and database parameters dialog box. [Figure 19 on page 206](#) shows a sample dialog box for these parameters.

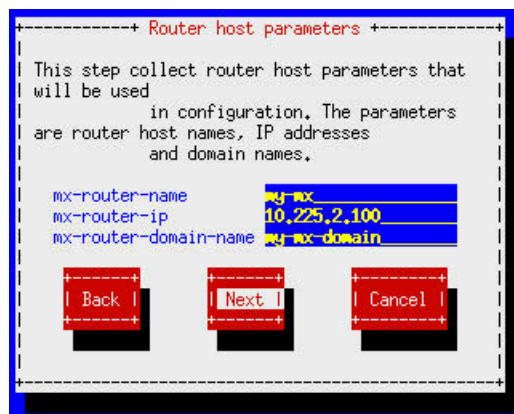
*Figure 19: SRC VTA Database Parameters Dialog Box*

```
+-----+ VTA database host and database parameters +-----+
|
| This step collect VTA database host and database
| parameters that will be used
|           in configuration. The parameters are
| VTA database host IP addresses, database
| connection user name and password.
|
| vta-database-ip    10.227.2.13
| vta-database-user   bla
| vta-database-pass   foo
|
| [Back] [Next] [Cancel]
|
+-----+
```

5. Enter the values for the Router host parameters dialog box. [Figure 20 on page 207](#) shows a sample dialog box for these parameters.



Figure 20: Router Host Parameters Dialog Box



6. Press Tab to highlight Finish and then press the Space bar to select it. The wizard displays a list of corresponding SRC CLI set commands reflecting the values you entered. Review the corresponding SRC CLI set commands. If you want to make changes, use Back until you reach the dialog box you want to change.
7. After you complete the configuration, select Commit from the set commands dialog box. The wizard responds with:

```

user@host# configuration-wizard wizard-name fair-usage-mx.wiz.xml
Please wait, it may take some minutes ...
Committed

[edit]
user@host#

```

#### Related Documentation

- [Fair Usage on MX Series Routers Configuration Wizard Overview on page 193](#)
- [Fair Usage on MX Series Routers Configuration Wizard Configuration Overview on page 194](#)

