



SRC-PE Software

Services and Policies Guide

Release 3.1.x

Juniper Networks, Inc.

1194 North Mathilda Avenue
Sunnyvale, California 94089
USA

408-745-2000

www.juniper.net

Part Number: 530-028669-01, Revision 1

Juniper Networks, the Juniper Networks logo, JUNOS, NetScreen, ScreenOS, and Steel-Belted Radius are registered trademarks of Juniper Networks, Inc. in the United States and other countries. JUNOS is a trademark of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the 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.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

SRC-PE Software Services and Policies Guide

Release 3.1.x

Copyright © 2009, Juniper Networks, Inc.

All rights reserved. Printed in USA.

Writing: Linda Creed, Justine Kangas, Betty Lew, Helen Shaw,

Editing: Fran Mues

Illustration: Nathaniel Woodward

Cover Design: Edmonds Design

Revision History

13 February 2009— Revision 1

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

YEAR 2000 NOTICE

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

SOFTWARE LICENSE

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

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

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

END USER LICENSE AGREEMENT

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

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

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

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

- a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.
- b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.
- c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.
- d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.
- e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

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

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

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

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

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

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

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

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

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

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

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

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

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous

agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Abbreviated Table of Contents

	About This Guide	xxi
Part 1	Managing Services and Service Schedules	
Chapter 1	Managing Services (SRC CLI)	3
Chapter 2	Managing Service Schedules	35
Chapter 3	Scheduling Services (SRC CLI)	43
Part 2	Defining Policies to Manage Traffic	
Chapter 4	Policy Management Overview	59
Chapter 5	Overview of Using Local and Global Parameters	81
Chapter 6	Configuring Local and Global Parameters (SRC CLI)	101
Chapter 7	Configuring and Managing Policies (SRC CLI)	105
Chapter 8	Policy Examples (SRC CLI)	195
Part 3	Generating Policies by Specifying Parameters	
Chapter 9	Defining and Acquiring Values for Parameters	209
Part 4	Index	
	Index	245

Table of Contents

About This Guide xxi

SRC Guides and Release Notes	xxi
Audience	xxi
Documentation Conventions	xxi
Related Juniper Networks Documentation	xxiii
Obtaining Documentation	xxv
Documentation Feedback	xxv
Requesting Technical Support	xxv

Part 1

Managing Services and Service Schedules

Chapter 1

Managing Services (SRC CLI) 3

Overview of Services for the SRC Software	3
Automatic Service Activation	4
Enabling the Service Configuration on the SRC CLI	4
Before You Configure SRC Services	4
Adding a Normal Service (SRC CLI)	5
Setting Parameter Values for Services (SRC CLI)	8
Aggregating Services	10
Overview of SRC Aggregate Services	10
Fragment Services	10
Subscriber Reference Expressions for Fragment Services	11
Mandatory Services	11
Redundant Services	11
Aggregate Service Sessions	11
Session Activation	12
Session Deactivation	12
Session Monitoring	12
Service Activation	13
Before You Configure an Aggregate Service	14
How Parameters Are Passed from Aggregate Service to Fragment Service	15
Configuring Service Fragments for an Aggregate Service (SRC CLI)	15
Configuring Timers for Aggregate Services (SRC CLI)	17
Using Python Expressions in a Subscriber Reference Expression	18
Sharing Service Provisioning	20
Overview of Sharing Service Provisioning	20
Adding an Infrastructure Service (SRC CLI)	20

Extending Service Implementations with Script Services	21
Overview of SRC Script Services	21
Customizing Service Implementations	22
Writing Scripts for Script Services	22
Configuring Script Services (SRC CLI)	24
Restricting Simultaneous Activation of Services	26
Overview of Restricting Simultaneous Activation of Services	26
Restricting Simultaneous Activation of Persistent or Automatic Services	27
Adding a Mutex Group (SRC CLI)	27
Customizing Service Delivery with Scopes	28
Overview of Restricting and Customizing Services for Subscribers	28
Assigning Service Scopes to Multiple VRs and Subscribers	29
Defining Multiple Scopes for a Service	29
Example: Using Service Scopes to Deliver a Limited Set of Services to Organizations	30
Example: Using Service Scopes to Customize Generic Services to Particular Regions	30
Configuring Service Scopes (SRC CLI)	32
Adding Service Scopes (SRC CLI)	32
Assigning Services and Mutex Groups to Service Scopes (SRC CLI)	32
Assigning Service Scopes to VRs or Subscribers (SRC CLI)	33
Restricting Service Activation	33

Chapter 2**Managing Service Schedules****35**

Overview of Service Schedules	35
Event-Based Schedules	35
Action Threshold	36
Preparation Time	36
Authorization Schedules	37
State-Based Schedules	37
Effective Period for Service Activation or Deactivation	38
One-Time Events and Recurring Events	39
Schedule Availability to Subscribers	39
Schedule Exclusions	39
Schedule Configuration Guidelines	40
Planning Service Schedules	41

Chapter 3**Scheduling Services (SRC CLI)****43**

Setting the Action Threshold and Preparation Time (SRC CLI)	43
Authorizing Scheduled Services (SRC CLI)	44
Adding a Service Schedule (SRC CLI)	45
Setting the Time Schedule (SRC CLI)	47
Guidelines for Entering Time Values for Service Schedules	49

Setting the Action for a Service Schedule (SRC CLI)	50
Defining Attributes for Service Activation (SRC CLI)	51
Example: Configuring Different Service Tiers for Different Days with the CLI	51
Example: Configuring a Service to Be Active During Nonwork Hours with the CLI	53
Example: Configuring a Service to Be Available for a Specified Interval with the CLI	55

Part 2

Defining Policies to Manage Traffic

Chapter 4

Policy Management Overview 59

Policy Management Overview	59
Policy Components	60
Policies, Services, and Subscribers CLI and the Policies, Services, and Subscribers Subtasks in the C-Web Interface	61
Policy Engine	61
Policy Repository	62
Policy Enforcement Point	62
Policy Information Model	62
Policy Objects	63
Policy Rules	64
Supported Conditions and Actions	65
Policy Conditions	66
Multiple Classifiers	67
Rate-Limiting with Multiple Classifiers	67
Expanded Classifiers	67
Policy Actions	68
Combining Actions	70
Dynamic Bandwidth Allocation Among Traffic Flows in SRC Policy Lists for JUNOS Routing Platforms	71
Overview of Rate-Limit Hierarchies	71
Parent Groups in Rate-Limit Hierarchies	71
Actions in Rate-Limit Hierarchies	72
Delivering QoS Services in a Cable Environment	73
Service Flow Scheduling Types	73
Client Type 1 Support	75
Proxied QoS with Policy Push	75
PCMM Gate	76
Session Class ID	76
PCMM Classifiers	76
PCMM Classifiers and Extended Classifiers	76
Guidelines for Configuring Classifiers	77
Traffic Profiles	77
DOCSIS Parameters	77
Service Class Name	78
FlowSpec Parameters	78

	Types of FlowSpec Services	78
	FlowSpec Parameters	79
	Marking Packets	79
Chapter 5	Overview of Using Local and Global Parameters	81
	Overview of Global and Local Parameters	81
	Parameter Types	82
	Predefined Global Parameters	90
	Naming Global Parameters	99
Chapter 6	Configuring Local and Global Parameters (SRC CLI)	101
	Viewing Predefined Global Parameters (SRC CLI)	101
	Configuring Global Parameters (SRC CLI)	101
	Configuring Local Parameters (SRC CLI)	103
	Viewing Runtime Parameters (SRC CLI)	104
Chapter 7	Configuring and Managing Policies (SRC CLI)	105
	Before You Configure SRC Policies	105
	Creating a Worksheet	105
	Naming Objects	106
	Using the apply-groups Statement	106
	Using Expressions	106
	Policy Values	106
	SAE to JUNOS Routing Platforms	106
	SAE to JUNOSe Routers	107
	Enabling the Policy Configuration on the SRC CLI	107
	Configuring Policy Folders (SRC CLI)	107
	Configuring Policy Groups (SRC CLI)	108
	Configuring Policy Lists (SRC CLI)	108
	Configuring Policy Rules	110
	Overview of Policy Rules	110
	Before You Configure JUNOS Policy Rules	110
	Setting the Policy Rule Precedence	111
	Adding a Policy Rule (SRC CLI)	111
	Classify-Traffic Conditions	113
	Configuring Classify-Traffic Conditions	113
	Before You Configure Classify-Traffic Conditions	115
	Enabling Expansion of JUNOSe Classify-Traffic Conditions (SRC CLI)	115
	Specifying the PCMM Classifier Type (SRC CLI)	116
	Specifying Port Access for Traffic Classification (SRC CLI)	117
	Creating a Classify-Traffic Condition (SRC CLI)	118
	Configuring Source Networks (SRC CLI)	119
	Configuring Source Grouped Networks (SRC CLI)	120
	Configuring Destination Networks (SRC CLI)	121
	Configuring Destination Grouped Networks (SRC CLI)	122
	Configuring Protocol Conditions (SRC CLI)	123

Configuring Protocol Conditions with Ports (SRC CLI)	124
Configuring Protocol Conditions with Parameters (SRC CLI)	127
Configuring TCP Conditions (SRC CLI)	131
Configuring ICMP Conditions (SRC CLI)	134
Configuring IGMP Conditions (SRC CLI)	135
Configuring IPsec Conditions (SRC CLI)	136
Configuring ToS Byte Conditions (SRC CLI)	138
Configuring JUNOS Filter Conditions (SRC CLI)	139
Configuring JUNOS Parent-Group References (SRC CLI)	140
Configuring JUNOS Secondary Input Policy Conditions (SRC CLI)	142
Configuring Application Protocol Conditions (SRC CLI)	144
Using Map Expressions in Application Protocol Conditions	147
Configuring QoS Conditions (SRC CLI)	147
Configuring Actions	148
Configuring Color Actions (SRC CLI)	149
Configuring Color Mark Actions (SRC CLI)	150
Configuring DOCSIS Actions (SRC CLI)	151
Configuring Exception Application Actions (SRC CLI)	156
Configuring Filter Actions (SRC CLI)	156
Configuring FlowSpec Actions (SRC CLI)	157
Configuring Forward Actions (SRC CLI)	159
Configuring Forwarding Class Actions (SRC CLI)	160
Configuring GateSpec Actions (SRC CLI)	161
Configuring HTTP Redirect Actions (SRC CLI)	162
Configuring Loss Priority Actions (SRC CLI)	163
Configuring Mark Actions (SRC CLI)	164
Configuring NAT Actions (SRC CLI)	165
Configuring Next-Hop Actions (SRC CLI)	166
Configuring Next-Interface Actions (SRC CLI)	167
Configuring Next-Rule Actions (SRC CLI)	168
Configuring Policer Actions (SRC CLI)	169
Configuring the Packet Action for the Policer Action (SRC CLI)	170
Configuring QoS Profile Attachment Actions (SRC CLI)	172
Configuring Rate-Limit Actions (SRC CLI)	173
Configuring Hierarchical Rate-Limit Actions (SRC CLI)	175
Configuring Types of Actions for Rate-Limit Actions (SRC CLI)	178
Configuring Reject Actions (SRC CLI)	180
Configuring Routing Instance Actions (SRC CLI)	181
Configuring Scheduler Actions (SRC CLI)	182
Configuring Drop Profiles (SRC CLI)	184
Configuring Service Class Name Actions (SRC CLI)	185
Configuring Stateful Firewall Actions (SRC CLI)	186
Configuring Template Activation Actions (SRC CLI)	188
Configuring Traffic-Class Actions (SRC CLI)	189
Configuring Traffic-Mirror Actions (SRC CLI)	190
Configuring Traffic-Shape Actions (SRC CLI)	191
Configuring User Packet Class Actions (SRC CLI)	192

Chapter 8 Policy Examples (SRC CLI) 195

Example: Creating Access Policies for Subscribers	195
Types of Policies	195
Sample Access Policies	195
DHCP Policy Group	196
PPP Policy Group	197
Example: Providing Tiered Internet Services with Policing	199
Types of Policies	199
Sample JUNOS Rate-Limiting Policy	199
Local Parameter	199
Policy List je-out	200
Policy List je-in	200
Sample JUNOS Policer Policy	201
Local Parameter	201
PolicyList j-out	201
PolicyList j-in	201
Defining the Tiered Internet Services	202
Internet-Gold Service	202
Internet-Silver Service	202
Internet-Bronze Service	203
Example: Providing Premium Services	203
Types of Policies	203
Sample JUNOS and JUNOS Content Provider Policies	204
PolicyList je-out	204
PolicyList j-out	204
PolicyList je-in	205
PolicyList j-in	205
Defining the Premium Services	205
Music Service	206
News Service	206

Part 3 Generating Policies by Specifying Parameters

Chapter 9 Defining and Acquiring Values for Parameters 209

Parameters and Substitutions	209
Value Acquisition for Single Subscriptions	210
Value Acquisition for Multiple Subscriptions	211
Defining Parameters for the SRC Software	212
Formatting Substitutions	214
Parameter Names and Types	215
Expressions in Parameters	215
Specifying Parameter Names	216
Formatting Numbers	217
Formatting Strings	217
Using IPv4 Addresses	217
Specifying Ranges	217

Formatting Lists	217
Formatting Maps	218
Using Keywords	218
Using Separators	218
Using Operators	218
Adding Comments to Substitutions	223
Validating Substitutions	224
Example: Parameter Value Substitution	224
Requirements	225
Overview	225
Types of Parameters	226
Parameter Configuration	226
Parameter Values After Value Acquisition	226
Configuration	228
Configuring the Default Value for a Global Parameter	228
Configuring a Policy Group	228
Configuring a Service	236
Creating an Enterprise Subscriber	238
Subscribing ABCInc to the GoldMetered Service	239

Part 4

Index

Index	245
-------------	-----

List of Figures

Figure 1: Sample Configuration of an Aggregate Service	10
Figure 2: Aggregate Service Activation	13
Figure 3: Sample Action Threshold	36
Figure 4: Sample Preparation Time	37
Figure 5: Sample Effective Period	38
Figure 6: Policy Management Components	61
Figure 7: Policy Object Organization	63
Figure 8: JUNOS Policy Rules with Supported Conditions and Actions	65
Figure 9: JUNOS Policy Rules with Supported Conditions and Actions	66
Figure 10: PCMM Policy Rules with Supported Conditions and Actions	66
Figure 11: Authorization Framework for Proxied QoS with Policy Push	75
Figure 12: Value Acquisition for Single Subscriptions	211
Figure 13: Value Acquisition for Multiple Subscriptions	212
Figure 14: Network Used in Parameter Substitution Example	225
Figure 15: Policies Applied to the Sample Network	227

List of Tables

Table 1: Notice Icons	xxii
Table 2: Text Conventions	xxii
Table 3: Juniper Networks C-series and SRC Technical Publications	xxiii
Table 4: Fields Used in Python Expressions for Aggregate Services	18
Table 5: Parameter Selection Example	29
Table 6: Schedule Availability to Service Subscribers	39
Table 7: Policy Actions	68
Table 8: DOCSIS Service Flow Scheduling Types	73
Table 9: Mapping FlowSpec Types	79
Table 10: Parameters Available for Each Type of Service	79
Table 11: Parameter Types	82
Table 12: Predefined Global Parameters	91
Table 13: Parameter Definitions	213
Table 14: Operations That You Can Use in Expressions	218

About This Guide

- SRC Guides and Release Notes on page xxi
- Audience on page xxi
- Documentation Conventions on page xxi
- Related Juniper Networks Documentation on page xxiii
- Obtaining Documentation on page xxv
- Documentation Feedback on page xxv
- Requesting Technical Support on page xxv

SRC Guides and Release Notes

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

Audience

This guide is intended for experienced system and network specialists working with JUNOS routers and JUNOS routing platforms in an Internet access environment. We assume that readers know how to use the routing platforms, directories, and RADIUS servers that they will deploy in their SRC networks.

If you are using the SRC software in a cable network environment, we assume that you are familiar with the PacketCable Multimedia Specification (PCMM) as defined by Cable Television Laboratories, Inc. (CableLabs) and with the Data-over-Cable Service Interface Specifications (DOCSIS) 1.1 protocol. We also assume that you are familiar with operating a multiple service operator (MSO) multimedia-managed IP network.

Documentation Conventions

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

Table 1: Notice Icons





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

Table 2: Text Conventions

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

Table 2: Text Conventions (continued)

Key names linked with a plus sign (+)	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
<i>Italic typeface</i>	<ul style="list-style-type: none"> ■ Emphasizes words. ■ Identifies book names. ■ Identifies distinguished names. ■ Identifies files, directories, and paths in text but not in command examples. 	<ul style="list-style-type: none"> ■ There are two levels of access: <i>user</i> and <i>privileged</i>. ■ <i>SRC-PE Getting Started Guide</i> ■ <i>o = Users, o = UMC</i> ■ The <i>/etc/default.properties</i> file.
Backslash	At the end of a line, indicates that the text wraps to the next line.	Plugin.radiusAcct-1.class = \net.juniper.srmt.sae.plugin\RadiusTrackingPluginEvent
Words separated by the symbol	Represent a choice to select one keyword or variable to the left or right of this symbol. (The keyword or variable may be either optional or required.)	diagnostic line

Related Juniper Networks Documentation

The most current SRC documentation is available at:

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

This Web site contains the documentation described in Table 3 on page xxiii.

A complete list of abbreviations used in this document set, along with their spelled-out terms, is provided in the *SRC-PE Getting Started Guide*.

Table 3: Juniper Networks C-series and SRC Technical Publications

Document	Description
Core Documentation Set	
<i>C2000 and C4000 Hardware Guide</i>	Describes the hardware platforms and how to install, maintain, replace, and troubleshoot them. The guide also includes specifications.
<i>C2000 and C4000 Quick Start Guide</i>	Describes how to get the C-series Controller up and running quickly. Intended for experienced installers who want to expedite the installation process.
<i>SRC-PE Getting Started Guide</i>	Describes the SRC software, how to set up an initial software configuration, how to integrate RADIUS servers, and how to upgrade the SRC software. It also explains how to manage a C-series Controller. The guide describes how to set up and start the SRC CLI and the C-Web interface, as well as other SRC configuration tools. It includes reference material for the SRC documentation.
<i>SRC-PE CLI User Guide</i>	Describes how to use the SRC CLI, configure and monitor the platform with the CLI, and control the CLI environment. The guide also describes how to manage SRC components with the CLI.

Table 3: Juniper Networks C-series and SRC Technical Publications *(continued)*

Document	Description
<i>SRC-PE Network Guide</i>	Describes how to use and configure the SAE, the NIC, the SRC-ACP (Admission Control Plug-In) application, and the External Subscriber Monitor application. This guide also provides detailed information about using JUNOSe routers, JUNOS routing platforms, and other network devices in the SRC network.
<i>SRC-PE Services and Policies Guide</i>	Describes how to work with services and policies. The guide provides an overview, configuration procedures, and management information. The guide also provides information about the SRC tools for configuring policies.
<i>SRC-PE Subscribers and Subscriptions Guide</i>	Describes how to work with residential and enterprise subscribers and subscriptions. The guide provides an overview, configuration procedures, and management information. This guide also provides information about the enterprise service portals, including the Enterprise Manager Portal.
<i>SRC-PE Monitoring and Troubleshooting Guide</i>	Describes how to use logging, the SNMP agent, the SRC CLI, and the C-Web interface to monitor and troubleshoot SRC components. This guide also describes the SNMP traps.
<i>SRC-PE Solutions Guide</i>	Provides high-level instructions for SRC implementations. The guide documents the following scenarios: managing QoS services on JUNOSe routers; managing subscribers in a wireless roaming environment; providing voice over IP (VoIP) services; integrating the SRC software in a PCMM environment, including the use of the Juniper Policy Server (JPS); and mirroring subscriber traffic on JUNOSe routers.
<i>SRC-PE CLI Command Reference, Volume 1</i> <i>SRC-PE CLI Command Reference, Volume 2</i>	Together constitute information about command and statement syntax; descriptions of commands, configuration statements, and options; editing level of statement options; and a history of when a command was added to the documentation.
<i>SRC PE NETCONF API Guide</i>	Describes how to use the NETCONF application programming interface (API) to configure or request information from the NETCONF server on a C-series Controller that runs the SRC software.
<i>SRC-PE XML API Configuration Reference</i>	Describes the tag elements in the SRC Extensible Markup Language (XML) application programming interface (API) that are equivalent to configuration statements in the SRC command-line interface (SRC CLI).
<i>SRC-PE XML API Operational Reference</i>	Describes the tag elements in the SRC Extensible Markup Language (XML) application programming interface (API) that are equivalent to operational commands in the SRC command-line interface (SRC CLI).
Application Library	
<i>SRC Application Library Guide</i>	Describes how to install and work with applications that you can use to extend the capabilities of the SRC software. The guide documents the following applications: SRC SOAP Gateway (SRC-SG) Web applications, an application to provide threat mitigation, an application to provide tracking and QoS control at the application level by integrating the SRC software with the Ellacoya deep packet inspection (DPI) platform, and an application to control volume usage.
Release Notes	

Table 3: Juniper Networks C-series and SRC Technical Publications (continued)

Document	Description
<i>SRC-PE Release Notes</i>	In the <i>Release Notes</i> , you will find the latest information about features, changes, known problems, resolved problems, supported platforms and network devices (such as Juniper Networks routers and CMTS devices), and third-party software. If the information in the <i>Release Notes</i> differs from the information found in the documentation set, follow the <i>Release Notes</i> .
<i>SRC Application Library Release Notes</i>	
	Release notes are included in the corresponding software distribution and are available on the Web.

Obtaining Documentation

To obtain the most current version of all Juniper Networks technical documents, see the products documentation page on the Juniper Networks Web site at <http://www.juniper.net/>.

To download complete sets of technical documentation to create your own documentation CD-ROMs or DVD-ROMs, see the CD-ROM and DVD-ROM Documentation page at

<http://www.juniper.net/techpubs/resources/cdrom.html>

Copies of the Management Information Bases (MIBs) are available at <http://www.juniper.net/>.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>. If you are using e-mail, be sure to include the following information with your comments:

- Document name
- Document part number
- Page number
- Software release version (not required for *Network Operations Guides [NOGs]*)

Requesting Technical Support

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

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the JTAC User Guide located at <http://www.juniper.net/customers/support/downloads/710059.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC Hours of Operation —The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

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

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

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

Opening a Case with JTAC

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

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

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

Part 1

Managing Services and Service Schedules

- Managing Services (SRC CLI) on page 3
- Managing Service Schedules on page 35
- Scheduling Services (SRC CLI) on page 43

Chapter 1

Managing Services (SRC CLI)

- Overview of Services for the SRC Software on page 3
- Enabling the Service Configuration on the SRC CLI on page 4
- Before You Configure SRC Services on page 4
- Adding a Normal Service (SRC CLI) on page 5
- Setting Parameter Values for Services (SRC CLI) on page 8
- Aggregating Services on page 10
- Sharing Service Provisioning on page 20
- Extending Service Implementations with Script Services on page 21
- Customizing Service Delivery with Scopes on page 28
- Configuring Service Scopes (SRC CLI) on page 32
- Restricting Service Activation on page 33

Overview of Services for the SRC Software

The SRC software supports four types of services:

- Normal—Policy-based service.
- Aggregate—Group of services, handled as a unit.
- Infrastructure—Service that can be provisioned only once and then activated a number of times for one or more subscribers across network devices.
- Script—Custom service into which you can insert or reference a script that provisions policies on a number of systems across a network, including networks that contain devices that do not have supported device drivers.

Use aggregate and infrastructure services together to apply policies across JUNOSe routers and JUNOS routing platforms, and other systems that have a supported device driver.

Use script services to create customized service implementations, such as a service to configure firewall policies on a device that does not have a supported device driver—for example, a Juniper Networks NetScreen-5GT appliance.

Automatic Service Activation

You can configure a permanent service—a service that the SAE automatically activates when it starts a subscriber session for subscribers who use that service. A typical application of this feature is to automatically activate a particular video service for all subscribers associated with a particular retailer. You can allow subscribers to deactivate the service, or prohibit them from deactivating it, after the SAE has automatically activated it. To make a service permanent, set the **permanent** option in the service configuration.

- Related Topics**
- Overview of SRC Aggregate Services on page 10
 - Enabling the Service Configuration on the SRC CLI on page 4
 - Before You Configure SRC Services on page 4
 - Adding a Normal Service (SRC CLI) on page 5
 - Setting Parameter Values for Services (SRC CLI) on page 8
 - Restricting Service Activation on page 33

Enabling the Service Configuration on the SRC CLI

Before you can configure services with the SRC CLI, you must enable the policy, service, and subscriber editor on the SRC CLI. To do so:

In operational mode, enter the following command:

```
user@host> enable component editor
```

- Related Topics**
- Overview of Services for the SRC Software on page 3
 - Enabling the Service Configuration on the C-Web Interface
 - Before You Configure SRC Services on page 4
 - Adding a Normal Service (SRC CLI) on page 5
 - Setting Parameter Values for Services (SRC CLI) on page 8

Before You Configure SRC Services

Before you configure services:

- Plan the services that you want to make available to subscribers.
- Configure the policies for a service to use. For information about configuring policies, see “Configuring Policy Groups (SRC CLI)” on page 108 and “Configuring Policy Lists (SRC CLI)” on page 108.

- Related Topics**
- Overview of Services for the SRC Software on page 3
 - Before You Configure SRC Services on page 4
 - Enabling the Service Configuration on the SRC CLI on page 4

- Adding a Normal Service (SRC CLI) on page 5
- Setting Parameter Values for Services (SRC CLI) on page 8

Adding a Normal Service (SRC CLI)

Use the following configuration statements to add normal services to the global service scope:

```
services global service name {
  description description ;
  type (normal);
  category category ;
  url url ;
  policy-group policy-group ;
  authentication-required;
  authorization-plug-in [ authorization-plug-in... ];
  tracking-plug-in [ tracking-plug-in... ];
  session-timeout session-timeout ;
  idle-timeout idle-timeout ;
  accounting-interim-interval accounting-interim-interval ;
  radius-class radius-class ;
  status (inactive | active);
  activate-only;
  permanent;
  available;
  secret;
  shared-service-name shared-service-name ;
}
```

Use the following configuration statements to add normal services to a service scope:

```
services scope name service name {
  description description ;
  type (normal | aggregate | script | infrastructure);
  category category ;
  url url ;
  policy-group policy-group ;
  authentication-required;
  authorization-plug-in [ authorization-plug-in... ];
  tracking-plug-in [ tracking-plug-in... ];
  session-timeout session-timeout ;
  idle-timeout idle-timeout ;
  accounting-interim-interval accounting-interim-interval ;
  radius-class radius-class ;
  status (inactive | active);
  activate-only;
  permanent;
  available;
  secret;
  shared-service-name shared-service-name ;
}
```

To add a normal service:

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

```
user@host# edit services global service Video-Gold
```

2. (Optional) Enter a description for the service.

```
[edit services global service Video-Gold]
user@host# set description description
```

3. Configure the type of service.

```
[edit services global service Video-Gold]
user@host# set type normal
```

4. (Optional) Configure the category of the service for other SRC applications, such as portals.

```
[edit services global service Video-Gold]
user@host# set category category
```

5. (Optional) Configure the link used in SRC applications, such as portals.

```
[edit services global service Video-Gold]
user@host# set url url
```

6. (Optional) Configure the policy group that is applied when the service is activated.

```
[edit services global service Video-Gold]
user@host# set policy-group policy-group
```

7. (Optional) Enable authentication required for services activated by portals.

```
[edit services global service Video-Gold]
user@host# set authentication-required
```

8. (Optional) Configure the plug-in(s) used to authorize the service.

```
[edit services global service Video-Gold]
user@host# set authorization-plug-in [ authorization-plug-in... ];
```

9. (Optional) Configure the plug-in(s) used to collect accounting data for the service.

```
[edit services global service Video-Gold]
user@host# set tracking-plug-in [ tracking-plug-in... ]
```

10. (Optional) Configure the time after which the service session is deactivated.

```
[edit services global service Video-Gold]
user@host# set session-timeout session-timeout
```


11. (Optional) Configure the idle time after which the SAE deactivates service.

```
[edit services global service Video-Gold]
user@host# set idle-timeout idle-timeout
```

12. (Optional) Configure the time between interim accounting messages.

```
[edit services global service Video-Gold]
user@host# set accounting-interim-interval accounting-interim-interval
```

13. (Optional) Configure the value of the RADIUS class attribute in accounting messages.

```
[edit services global service Video-Gold]
user@host# set radius-class radius-class
```

14. (Optional) Configure the status of the service.

```
[edit services global service Video-Gold]
user@host# set status (inactive | active)
```

15. (Optional) Configure whether the SAE can deactivate this service

```
[edit services global service Video-Gold]
user@host# set activate-only
```

16. (Optional) Enable automatic activation when the service is subscribed.

```
[edit services global service Video-Gold]
user@host# set permanent
```

17. (Optional) Specify whether a subscriber can activate a service.

```
[edit services global service Video-Gold]
user@host# set available
```

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

```
[edit services global service Video-Gold]
user@host# set secret
```

19. (Optional) Verify your configuration.

```
[edit services global service Video-Gold]
user@host# show
description "Example for content provider allowing high speed access";
type normal;
category Video;
url http://video.server.com;
policy-group /sample/common/content-provider-tiered;
radius-class Video-Gold;
```

```
status active;
}
```

- Related Topics**
- Overview of Services for the SRC Software on page 3
 - Adding a Normal Service (C-Web Interface)
 - Adding an Infrastructure Service (SRC CLI) on page 20
 - Setting Parameter Values for Services (SRC CLI) on page 8

Setting Parameter Values for Services (SRC CLI)

Using parameters, you can define general settings in one SRC object and provide specific values for that setting in another object. For example, you can define the general settings for a rate limiter in a policy, insert a parameter for a rate in the policy, and provide specific values for the rate in each service that uses this policy. For information about the concept of parameters, see “Parameters and Substitutions” on page 209.

Use the following configuration statements to configure parameters for services in the global service scope:

```
services global service name parameter {
  gateway-ip-address gateway-ip-address ;
  service-ip-address service-ip-address ;
  service-ip-mask service-ip-mask ;
  service-port service-port ;
  substitution [ substitution... ];
  session-volume-quota session-volume-quota ;
}
```

Use the following configuration statements to configure parameters for services in a service scope:

```
services scope name service name parameter {
  gateway-ip-address gateway-ip-address ;
  service-ip-address service-ip-address ;
  service-ip-mask service-ip-mask ;
  service-port service-port ;
  substitution [ substitution... ];
  session-volume-quota session-volume-quota ;
}
```

To configure parameters for services:

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

```
user@host# edit services global service Video-Gold parameter
```

2. (Optional) Configure the actual IP address of the gateway router. This value is substituted for the policy global parameter called `gateway_ipAddress`.

```
[edit services global service Video-Gold parameter]
user@host# set gateway-ip-address gateway-ip-address
```

3. (Optional) Configure the actual IP address of the host(s) that provides the service. This value is substituted for the policy global parameter called `service_ipAddress`.

```
[edit services global service Video-Gold parameter]
user@host# set service-ip-address service-ip-address
```

4. (Optional) Configure the actual IP mask for the service. This value is substituted for the policy global parameter called `service_ipMask`.

```
[edit services global service Video-Gold parameter]
user@host# set service-ip-mask service-ip-mask
```

5. (Optional) Configure the actual port for the service. This value is substituted for the policy global parameter called `service_port`.

```
[edit services global service Video-Gold parameter]
user@host# set service-port service-port
```

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

```
[edit services global service Video-Gold parameter]
user@host# set substitution [ substitution... ]
```

7. (Optional) Configure the quota for the volume of data for service sessions. The SRC software uses this value as the default for service sessions created for this service.

```
[edit services global service Video-Gold parameter]
user@host# set session-volume-quota session-volume-quota
```

8. (Optional) Verify your configuration.

```
[edit services global service Video-Gold parameter]
user@host# show
service-ip-address 10.10.40.16;
service-ip-mask 255.255.255.240;
substitution "bw = 5000000";
```

- Related Topics**
- Overview of Services for the SRC Software on page 3
 - Setting Parameter Values for Services (C-Web Interface)
 - Enabling the Service Configuration on the SRC CLI on page 4
 - Adding a Normal Service (SRC CLI) on page 5

Aggregating Services

An aggregate service comprises a number of individual services. Topics include:

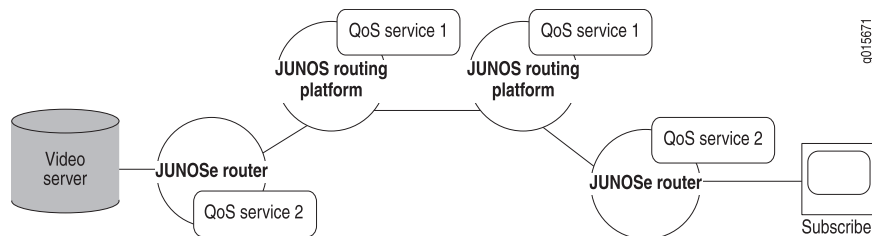
- Overview of SRC Aggregate Services on page 10
- Before You Configure an Aggregate Service on page 14
- How Parameters Are Passed from Aggregate Service to Fragment Service on page 15
- Configuring Service Fragments for an Aggregate Service (SRC CLI) on page 15
- Configuring Timers for Aggregate Services (SRC CLI) on page 17
- Using Python Expressions in a Subscriber Reference Expression on page 18

Overview of SRC Aggregate Services

Combining services lets the SRC software treat the services within an aggregate service as a unit. When an aggregate service becomes active, it tries to activate all the services within it.

An aggregate service can distribute the activation of a number of services within the aggregate across one or more SAEs in an SRC network. This specialized service is ideal for supporting voice over IP (VoIP) and video on demand. To deliver these types of features to subscribers, you can configure bidirectional or unidirectional quality of service (QoS) services based on policies provisioned across a number of interfaces on one or more SAE-managed network devices in an SRC network. Figure 1 on page 10 shows a sample aggregate service that provides end-to-end QoS for video on demand, with QoS service 1 and QoS service 2 activated on Juniper Networks routers in the path between the video server and the subscriber.

Figure 1: Sample Configuration of an Aggregate Service



The services included in an aggregate service manage policies in the usual manner. The aggregate service does not directly manage any policies on a network device.

Fragment Services

The services that make up an aggregate service are referred to as fragment services. This term provides a way to distinguish between services that are included in an aggregate service and those that are not. The fragment services can be any type of service that the SAE supports, except another aggregate service.

Subscriber Reference Expressions for Fragment Services

The configuration for each fragment service includes a subscriber reference expression, a phrase that identifies the subscriber sessions that activate the fragment service. The subscriber reference expression defines the subscriber session by subscriber IP address, distinguished name (DN), interface name, login name, or associated virtual router.

To use aggregate services requires that the network information collector (NIC) be configured. Use a configuration scenario that provides a key for the type of subscriber reference expression defined for the fragment service. For example, if the subscriber reference expression is a DN, the NIC key is also a DN. In this case, you could use the NIC configuration scenario OnePopDnSharedIp, which uses a DN as a key.

For more information about the NIC configuration scenarios and the types of resolutions performed by these scenarios, see NIC Configuration Scenarios.

Mandatory Services

A fragment service that must be active for an aggregate service to become active is called a mandatory service. When you configure an aggregate service, you specify which services, if any, are mandatory. For example, you could specify that rate-limiting services for a video-on-demand connection be mandatory to ensure call quality.

Redundant Services

When you configure an aggregate service, you can configure fragment services to provide redundancy for each other. Fragment services that share the same redundancy group name provide redundancy.

For an aggregate service to become active, at least one fragment service from each redundancy group must become active. For example, if you configure two services, S1 and S2, and assign the same redundancy group name to each of these services, S1 and S2 provide redundancy for each other if one becomes disabled.

While an aggregate service is active, the SAE tries to keep all fragment services within it active. An aggregate service and any of its active fragment services become inactive if a mandatory fragment service or an entire redundancy group becomes inactive.

Aggregate Service Sessions

An aggregate service session coordinates the activation of the services within it. It runs on the same SAE where it starts. The aggregate service session is created in the router driver that hosts the subscriber session that starts the service. An individual service session for a fragment service can be activated in the same SAE or another SAE on the SRC network.

Understanding how aggregate service sessions are managed can help you troubleshoot service activation or service deactivation issues that might arise. The SRC software provides a set of configurable timers that helps control session management.

For information about the timers that you can use to troubleshoot aggregate services, see “Configuring Timers for Aggregate Services (SRC CLI)” on page 17 .

Session Activation

An aggregate service becomes active when:

- All mandatory services are active.
If a mandatory service does not start, the SAE deactivates any fragment services that are active.
- If there are no mandatory services, at least one service is active.

If any fragment services that are not mandatory services do not become active, the aggregate service continues to try to start them. How long the aggregate service tries to activate fragment services depends on the settings for activation-deactivation time.

When an aggregate service becomes active, it monitors the services that are part of the aggregate service.



NOTE: Depending on your implementation, accounting software could detect that a fragment service session became active even though the associated aggregate service did not become active, resulting in the fragment services being deactivated.

You can configure your accounting software to ignore the activation of the fragment session when an aggregate service session fails. This way, a customer is not billed for an aggregate service that was not received.

Session Deactivation

When the SAE deactivates an aggregate service, the aggregate service session tries to deactivate the services within it. The SAE deactivates an aggregate service when all fragment services stop. If one of these services remains active, the aggregate service stays in memory until the service session ends. The SAE periodically tries to stop the active fragment session until the maximum retry time is reached, at which time it deactivates the aggregate service. As a result, the aggregate service session can remain in memory after the associated subscriber session ends.

Session Monitoring

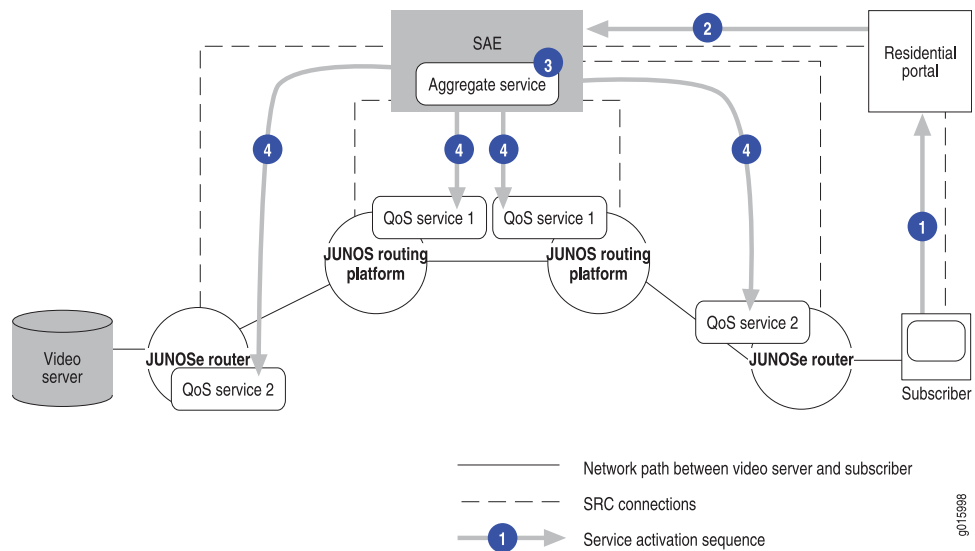
An aggregate service session exchanges keepalive messages with a session management process for remote fragment services. This way, if a service session is removed from a router while the SAE is not managing the router, such as when the Common Open Policy Service (COPS) client stops on a JUNOS router or the configuration database is reset on a JUNOS routing platform, the SAE associated with the router receives notification that the keepalive message failed.

Service Activation

Aggregate services are activated in a way similar to any other service, but with the additional requirement of activating the associated fragment services.

Figure 2 on page 13 shows a sample service activation for a video-on-demand service.

Figure 2: Aggregate Service Activation



The following process describes the service activation for a video-on-demand service, with Steps 1–4 illustrated in Figure 2 on page 13.

1. A subscriber requests a video-on-demand service through a residential portal.
2. The residential portal requests the service through the SAE.
3. The SAE activates a subscription for the associated aggregate service, and a session for the aggregate service becomes active.
4. The aggregate service coordinates with the SAE, and the SAE tries to activate the fragment services that have been configured for the aggregate service.

The aggregate service becomes active when:

- All mandatory services are active.
- If there are no mandatory services, at least one fragment service is active.
- For redundant fragment services, at least one fragment service configured for a redundancy group becomes active.

The aggregate service initiates accounting, if accounting has been configured.

After the aggregate service becomes active, it monitors fragment services to ensure that they are still active. When the subscriber or the video server ends the video-on-demand session, the aggregate service tries to terminate active fragment services.

- Related Topics**
- How Parameters Are Passed from Aggregate Service to Fragment Service on page 15
 - Overview of Services for the SRC Software on page 3
 - Login Events
 - Before You Configure an Aggregate Service on page 14
 - Configuring Service Fragments for an Aggregate Service (SRC CLI) on page 15
 - Using Python Expressions in a Subscriber Reference Expression on page 18
 - Defining Attributes for Service Activation (SRC CLI) on page 51

Before You Configure an Aggregate Service

Before you configure an aggregate service:

1. Plan the aggregate service:
 - Plan which fragment services will constitute the aggregate service.
 - Plan the routers on which the fragment services are to be activated.
 2. Configure the fragment services.
- See “Adding a Normal Service (SRC CLI)” on page 5 .
3. If the aggregate service includes services to be activated remotely, ensure that one or more NIC proxies are configured on each SAE.
 4. Ensure that the NIC is configured to use a scenario that provides the appropriate type of key.

See NIC Configuration Scenarios.

5. Ensure that the SAEs can communicate with each other and the NIC host(s). Make sure that firewalls permit TCP and CORBA communication between the systems hosting the SAEs, and communication between the NIC host(s) and the SAE.

See Port Settings for SRC Components.

6. Ensure that the communication between SAEs is secure.

Follow the standards for your organization to ensure that communication between SAEs is protected.

7. If the aggregate service is to include a fragment service on a remote SAE, ensure that the remote fragment service can become active by verifying that the fragment service is loaded on the remote SAE.

- Related Topics**
- Overview of SRC Aggregate Services on page 10
 - How Parameters Are Passed from Aggregate Service to Fragment Service on page 15
 - Configuring Service Fragments for an Aggregate Service (SRC CLI) on page 15

- Configuring Timers for Aggregate Services (SRC CLI) on page 17
- Defining Attributes for Service Activation (SRC CLI) on page 51

How Parameters Are Passed from Aggregate Service to Fragment Service

There are two ways to set up parameters in aggregate and fragment services:

- If you use just a parameter name in the aggregate service, for example `user_IpAddress`, then the value of `user_IpAddress` in the aggregate session is bound to the name `user_IpAddress` in the fragment service.
- If you use `user_IpAddress` as the parameter name and `fragSubrIp = user_IpAddress` as a substitution in the aggregate service, `user_IpAddress` is given a different name in the fragment service session. The parameter name `fragSubrIps` in the fragment service session is bound to the value of `user_IpAddress` in the aggregate service session.

Use this scheme to configure parameters and substitutions when the parameter in the aggregate service session has a name that is already used in the fragment for something else. A common example is `user_IpAddress`, which is usually defined in all service sessions. This scheme is also useful when you are aggregating services developed independently. You can call the aggregate service parameters whatever makes sense in that context, and name the fragment service parameters independently.

- Related Topics**
- Overview of SRC Aggregate Services on page 10
 - Configuring Service Fragments for an Aggregate Service (SRC CLI) on page 15
 - Configuring Timers for Aggregate Services (SRC CLI) on page 17
 - Using Python Expressions in a Subscriber Reference Expression on page 18

Configuring Service Fragments for an Aggregate Service (SRC CLI)

Use the following configuration statements to configure an aggregate service in the global service scope:

```
services global service name aggregate fragment name {
    expression expression ;
    service service ;
    mandatory;
    redundancy-group redundancy-group ;
    subscription-required;
    substitution [ substitution... ];
}
```

Use the following configuration statements to configure an aggregate service in a service scope:

```
services scope name service name aggregate fragment name {
    expression expression ;
    service service ;
}
```

```

mandatory;
redundancy-group redundancy-group ;
subscription-required;
substitution [ substitution... ];
}

```

To configure service fragments for an aggregate service:

1. From configuration mode, enter the service aggregate configuration. In this sample procedure, the service called MirrorAggregate is configured in the scope configuration.

```

user@host# edit services scope TM service MirrorAggregate aggregate fragment 0

```

2. Configure the subscriber reference expression that identifies the remote subscriber session that will host the fragment.

```

[edit services scope TM service MirrorAggregate aggregate fragment 0]
user@host# set expression expression

```

3. Configure the name of the service to be included in the aggregate service as a fragment service.

```

[edit services scope TM service MirrorAggregate aggregate fragment 0]
user@host# set service service

```

4. (Optional) Specify whether the fragment service must be active for the aggregate service to become active.

```

[edit services scope TM service MirrorAggregate aggregate fragment 0]
user@host# set mandatory

```

5. (Optional) Configure the group name to be applied to each fragment service that is to be part of a redundancy group.

```

[edit services scope TM service MirrorAggregate aggregate fragment 0]
user@host# set redundancy-group redundancy-group

```

6. (Optional) Specify whether a remote subscriber session is required to subscribe to the fragment service.

```

[edit services scope TM service MirrorAggregate aggregate fragment 0]
user@host# set subscription-required

```

7. (Optional) Configure the list of substitutions that are used as arguments for the fragment to become active.

```

[edit services scope TM service MirrorAggregate aggregate fragment 0]
user@host# set substitution [ substitution... ]

```

8. (Optional) Verify your configuration.

```
[edit services scope TM service MirrorAggregate aggregate fragment 0]
user@host# show
expression
"vr=\"<- substitution.vrNames ->\",
interfaceName=\"FORWARDING_INTERFACE\"";
service MirrorFragment;
substitution fragSubrIps=subrIps;
```

- Related Topics**
- Overview of SRC Aggregate Services on page 10
 - How Parameters Are Passed from Aggregate Service to Fragment Service on page 15
 - Configuring Service Fragments for an Aggregate Service (C-Web Interface)
 - Using Python Expressions in a Subscriber Reference Expression on page 18

Configuring Timers for Aggregate Services (SRC CLI)

You can change the values for several timers to specify the intervals associated with monitoring and activating aggregate sessions. Use the following configuration statements to configure these timers and intervals:

```
shared sae configuration aggregate-services {
  keepalive-time keepalive-time ;
  keepalive-retry-time keepalive-retry-time ;
  activation-deactivation-time activation-deactivation-time ;
  failed-notification-retry-time failed-notification-retry-time ;
}
```

To configure timers used by aggregate services:

1. From configuration mode, enter the shared sae aggregate service configuration.

```
user@host# edit shared sae configuration aggregate-services
```

2. Configure the interval at which keepalive messages are sent between an aggregate service session and an associated remote service management session to verify that an aggregate service is active.

```
[edit shared sae configuration aggregate-services]
user@host# set keepalive-time keepalive-time
```

3. Configure the time to wait for an acknowledgement of a keepalive message before sending a new keepalive message if a response to a keepalive message is not received.

```
[edit shared sae configuration aggregate-services]
user@host# set keepalive-retry-time keepalive-retry-time
```

4. Configure the length of time to continue to try to activate or deactivate a fragment service session.

```
[edit shared sae configuration aggregate-services]
user@host# set activation-deactivation-time activation-deactivation-time
```

5. Configure the length of time to continue sending failure notifications if an aggregate service cannot reach a fragment service, or a fragment service cannot reach an aggregate service during shutdown of the aggregate service.

```
[edit shared sae configuration aggregate-services]
user@host# set failed-notification-retry-time failed-notification-retry-time
```

6. (Optional) Verify your configuration.

```
[edit shared sae configuration aggregate-services]
user@host# show
keepalive-time 150000;
keepalive-retry-time 900;
activation-deactivation-time 900;
failed-notification-retry-time 9200;
```

- Related Topics**
- Overview of SRC Aggregate Services on page 10
 - How Parameters Are Passed from Aggregate Service to Fragment Service on page 15
 - Configuring Timers for Aggregate Services (C-Web Interface)
 - Configuring Service Fragments for an Aggregate Service (SRC CLI) on page 15

Using Python Expressions in a Subscriber Reference Expression

You can compose Python expressions from one or more of the fields in Table 4 on page 18 for the definition of a subscriber reference expression of a fragment service. You enter these expressions with the **expression** option of the **services scope *name* service *name* aggregate fragment** or **edit services global service *name* aggregate fragment** statement.

Table 4: Fields Used in Python Expressions for Aggregate Services

Field	Description
substitution. <xyz>	Value of the parameter <xyz> .
	Substitutions are acquired by means of the regular acquisition path for service sessions.
	The names of parameters are restricted to valid Python identifiers, such as 'ALPHA/' _" *(ALPHA/ DIGIT/" _")', with the exception of keywords, such as for , if , while , return , and , or , not , def , class , try , except . For the full list of Python keywords, see http://docs.python.org/ref/keywords.html .

Table 4: Fields Used in Python Expressions for Aggregate Services *(continued)*

Field	Description
loginType	<p>The type of subscriber session, one of the following:</p> <ul style="list-style-type: none"> ■ ASSIGNEDIP—An assigned IP login is triggered when an application accesses a subscriber object for an assigned IP subscriber that is not currently loaded into memory. (JUNOSe routers) ■ AUTHINTF—An authenticated interface login is triggered when an interface responds to authentication, such as authentication for a PPP session. (JUNOSe routers) ■ INTF—An interface login is triggered when an interface comes up and the interface classifier script determines that the SAE should manage that interface, unless the interface comes up as a result of an authenticated PPP session. (JUNOS routing platforms and JUNOSe routers) ■ ADDR—An address login is triggered when the DHCP server in the JUNOSe router provides a token IP address. (JUNOSe routers) ■ AUTHADDR—An authenticated address login is triggered when the DHCP server in the JUNOSe router provides a public IP address. (JUNOSe routers) ■ PORTAL—A portal login is triggered when the portal API is invoked by a JSP Web page to log in a subscriber. (JUNOS routing platforms and JUNOSe routers)
loginName	Login name provided by a subscriber
userName	Username portion of the loginName
domainName	Domain name portion of the loginName
serviceBundle	Content of the vendor-specific RADIUS attribute for service bundle
radiusClass	RADIUS class used for authorization
virtualRouterName	Name of virtual router in the format vrname@hostname
interfaceName	Name of the interface
ifAlias	Description of the interface configured on the router
ifDesc	<p>Alternate name for the interface. This is the name used by the Simple Network Management Protocol (SNMP).</p> <p>On a JUNOSe router the format of the description is:</p> <p>ip < slot > / < port > . < subinterface ></p> <p>On a JUNOS routing platform, ifDesc is the same as interfaceName.</p>
nasPortId	Port identifier of an interface, including the interface name and additional layer 2 information (for example, fastEthernet 3/1)
macAddress	Text representation of the MAC address for the DHCP subscriber (for example, 00:11:22:33:44:55)
retailerDn	Distinguished name of the retailer

Table 4: Fields Used in Python Expressions for Aggregate Services *(continued)*

Field	Description
nasIp	Network access server IP address of the router
dhcp	DHCP options. See Overview of Classification Scripts .
primaryUserName	PPP or DHCP username. This name does not change when the subscriber logs in through a portal.

- Related Topics**
- Overview of Restricting and Customizing Services for Subscribers on page 28
 - Overview of SRC Aggregate Services on page 10
 - Configuring Service Fragments for an Aggregate Service (SRC CLI) on page 15

Sharing Service Provisioning

- Overview of Sharing Service Provisioning on page 20
- Adding an Infrastructure Service (SRC CLI) on page 20

Overview of Sharing Service Provisioning

You can use infrastructure services to provision a service to be shared by a number of subscriber sessions. Infrastructure services are services that can be activated a number of times for one or more subscribers, but provisioned only once. Infrastructure services are designed to be shared among instances of aggregate services.

When an infrastructure service is activated, the SAE activates the service if a shared service session for the service is not already active; otherwise, it increments the usage counter for the service. When an infrastructure service is deactivated, the SAE decrements the usage counter for the shared session. When the last service session is deactivated, the shared session is also deactivated.

Although an infrastructure service is designed for use as a fragment service in an aggregate service, it can be used independently. As a fragment service, it can be bundled with other fragment services to deliver a service package in the aggregate service.

- Related Topics**
- Overview of SRC Script Services on page 21
 - Overview of SRC Aggregate Services on page 10
 - Adding an Infrastructure Service (SRC CLI) on page 20

Adding an Infrastructure Service (SRC CLI)

To add an infrastructure service:

1. Add the service to be shared, as described in “Adding a Normal Service (SRC CLI)” on page 5.

2. Set the service type to infrastructure.

```
[edit services global service Infrastructure]
user@host# set type infrastructure
```

3. Configure the name of the service to be shared.

```
[edit services global service Infrastructure]
user@host# set shared-service-name shared-service-name
```

4. (Optional) Verify your configuration.

```
[edit services global service Infrastructure]
user@host# show
type infrastructure;
radius-class infrastructure;
status active;
shared-service-name Video-Bronze;
```

- Related Topics**
- Overview of Sharing Service Provisioning on page 20
 - Customizing Service Implementations on page 22
 - Adding an Infrastructure Service (C-Web Interface)

Extending Service Implementations with Script Services

- Overview of SRC Script Services on page 21
- Customizing Service Implementations on page 22
- Restricting Simultaneous Activation of Services on page 26

Overview of SRC Script Services

You use services to provision policies on a number of systems across a network, including networks that do not contain a JUNOS router or JUNOS routing platform. Script services provide an interface to call scripts that supply custom services. You can use script services to create custom service implementations, such as:

- Provisioning layer 2 devices, such as digital subscriber line access multiplexers (DSLAMs).
- Setting up network connections, such as MPLS tunnels.
- Provisioning policies for network devices that do not have a supported SAE device driver.
- Accessing functionality not currently supported by SAE device drivers but supported by other interfaces, such as RADIUS change of authorization (CoA) on JUNOS routers and JunoS provisioning on JUNOS routing platforms.

- Related Topics**
- Adding a Mutex Group (SRC CLI) on page 27
 - Customizing Service Implementations on page 22

Customizing Service Implementations

Perform the following tasks to customize service implementations:

1. Writing Scripts for Script Services on page 22
2. Configuring Script Services (SRC CLI) on page 24

Writing Scripts for Script Services

The ScriptService service provider interface (SPI) provides a Java interface that a script service implements. For information about the ScriptService interface and the ServiceSessionInfo interface, see the script service documentation in the SAE core API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>.

The implementation of the ScriptService interface activates the service. The SAE sends authentication and tracking events when it activates, modifies, or deactivates a script service session.

The SAE supports script services written in Java or Jython. For scripts written in Java, you must compile and package the implemented ScriptService to make it available for use by the SAE. A Java implementation can include more than one Java archive (JAR) file.

The SAE synchronizes methods used by the same instance of the ScriptService class. You do not need to provide synchronized implementation of the methods.



NOTE: The script service implementation can be called by different threads at the same time. If your script uses resources that are shared between different service instances, you are responsible for synchronizing access to those resources.

To write a script to be used by a script service:

1. Create a class that provides a default constructor and that implements the ScriptService interface.
2. Manage activation and manipulation of the service session by implementing the following ScriptService methods:
 - activateSession()—Activates the script service session.
 - deactivateSession()—Deactivates the script service session and returns any final accounting data for the script service session.
 - modifySession()—If the counters were reset during the modification, modifies the script service session and returns any accounting data.

These methods are implemented by the script service. They perform the associated action (activate, deactivate, modify) when the SAE calls the method.

3. (Optional) Get information about service sessions by using methods on the `ServiceSessionInfo` interface.
4. (Optional) Provide accounting data, if used, by using the following `ScriptService` method:

`getAccountingData()`—Polls for current accounting data and returns any current accounting data.

5. (Optional) Provide service status information by using the following `ScriptService` method:

`getState()`—Returns session data to be stored persistently on the router. The SAE does not use this data but provides it to the script when a service session is restored after failover.

6. Manage the script service by using the following `ScriptService` methods:
 - `initState()` —Initializes a recovered script service session after a state synchronization.
 - `discarded()`—Provides notification that the service session has been discarded. Service sessions are discarded when the SAE loses connection to a router. A discarded service session continues to exist on the router and is restored after the connection to the router is reestablished by an SAE.

The script service session releases any resources associated with a discarded session, but must not take any action to disrupt the service session.

You can also use the `stopService()` method on the `ServiceSessionInfo` object to stop a service and remove the service from the SAE. For example, consider a script service that monitors a state that it creates outside the SAE. If the script detects that the service is not active, it can stop the service and remove it from the SAE. You could use this type of script service to start a daemon process and monitor the process to make sure that it is alive.



NOTE: The `ScriptService` SPI does not provide access to a router driver.

**Example: Using the
ScriptService SPI in
Jython**

The following example implements the `ScriptService` SPI in Jython.

```
from net.juniper.smgt.sae.scriptservice import ScriptService
class SampleService(ScriptService):
    def initSessionInfo(self, ssi):
        self.ssi = ssi
    def activateSession(self):
        print "Activating ServiceName %s" % self.ssi.serviceName
    def deactivateSession(self):
        print "Deactivating ServiceName %s" % self.ssi.serviceName
        return None
    def modifySession(self, ssi):
        self.ssi = ssi
```

```

print "Modifying ServiceName %s" % self.ssi.serviceName
return None
def getAccountingData(self):
print "Getting accounting data for ServiceName %s" % self.ssi.serviceName
return None
def getState(self):
return None
def initState(self, ssi, state):
self.ssi = ssi
pass
def discarded(self):
pass

```

Example: Using the ScriptService SPI in Java

The following example implements the ScriptService SPI in Java.

```

class SampleService implements ScriptService {
    private ServiceSessionInfo ssi;
    public SampleService() { }
    public void initSessionInfo(ServiceSessionInfo ssi) {
        this.ssi = ssi;
    }
    public void activateSession() {
        System.out.println(" Activating ServiceName " +ssi.getServiceName());
    }
    public AccountingData deactivateSession() {
        System.out.println(" Deactivating ServiceName " +ssi.getServiceName());
        return null;
    }
    public AccountingData modifySessionSession(ServiceSessionInfo ssi) {
        this.ssi = ssi;
        System.out.println(" Modifying ServiceName " +ssi.getServiceName());
        return null;
    }
    public AccountingData getAccountingData() {
        System.out.println(" Getting accounting data for ServiceName "
+ssi.getServiceName());
        return null;
    }
    public byte[] getState() {
        return null;
    }
    public initState(ServiceSessionInfo ssi, byte[] state) {
        this.ssi = ssi;
    }
    public void discarded() {
    }
}

```

Configuring Script Services (SRC CLI)

Before you configure a script service, make sure that you know the location of the script file that the service will reference.

Use the following configuration statements to configure a script service in the global service scope:

```

services global service name script {
    script-type (url | python | java-class | java-archive);
    class-name class-name;

```

```

    file file ;
    filename filename;
}

```

Use the following configuration statements to configure a script service in a service scope:

```

services scope name service name script {
    script-type (url | python | java-class | java-archive);
    class-name class-name;
    file file ;
    filename filename;
}

```

To configure a script service:

1. Configure a normal service, but set the service type to **script**. See “Adding a Normal Service (SRC CLI)” on page 5 .
2. From configuration mode, enter the service script configuration. In this sample procedure, the service called `scriptService` is configured in the scope configuration.

```

user@host# edit services scope script service scriptService script

```

3. Configure the type of script that the script service uses.

```

[edit services scope script service scriptService script]
user@host# set script-type (url | python | java-class | java-archive)

```

4. For Java class and Python script services, configure the name of the class that implements the script service.

```

[edit services scope script service scriptService script]
user@host# set class-name class-name

```

5. Script services of the `url` script type are written in Java and provided as a Java archive (`.jar`) file. Configure the URL that specifies the location of the `.jar` file containing the script service implementation. For other script types, you must load the script service implementation.

```

[edit services scope script service scriptService script]
user@host# set file file

```

6. For Java class, and Java archive, and Python script services, load the script service implementation by specifying the filename of the local file containing the script service implementation. This file is the uncompiled Python source code or the compiled result of the Java script service (binary `.class` or `.jar` file).

```

[edit services scope script service scriptService script]
user@host# set filename filename

```

7. (Optional) Verify your configuration.

```
[edit services scope script service scriptService script]
user@host# show
script-type url;
class-name net.juniper.smgmt.script.Service;
file http://some-server/some-path/script-service.jar;
```

- Related Topics**
- Overview of SRC Script Services on page 21
 - Configuring Script Services (C-Web Interface)
 - Defining Attributes for Service Activation (SRC CLI) on page 51

Restricting Simultaneous Activation of Services

- Overview of Restricting Simultaneous Activation of Services on page 26
- Restricting Simultaneous Activation of Persistent or Automatic Services on page 27
- Adding a Mutex Group (SRC CLI) on page 27

Overview of Restricting Simultaneous Activation of Services

A mutex group defines a set of services that are mutually exclusive—services that the SAE cannot simultaneously activate for a particular subscriber. You can assign a service to more than one mutex group. When a subscriber requests activation of a particular service, the SAE determines which mutex groups contain that service. If the subscriber has current activations of other services listed in those mutex groups, the SAE proceeds in one of the following ways, depending on how you configured the mutex groups:

- Deactivates the other services listed in the mutex groups, and then activates the requested service.
- Refuses access to the requested service.

If the requested service is not listed in a mutex group, the SAE can activate the service regardless of any other services that the subscriber is using.

- Related Topics**
- Overview of Restricting and Customizing Services for Subscribers on page 28
 - Overview of SRC Aggregate Services on page 10
 - Restricting Service Activation on page 33
 - Restricting Simultaneous Activation of Persistent or Automatic Services on page 27
 - Adding a Mutex Group (SRC CLI) on page 27

Restricting Simultaneous Activation of Persistent or Automatic Services

The SAE uses the following method to prevent simultaneous activation of mutually exclusive services that are configured for persistent activation or that are activated automatically when a subscriber logs in:

1. If you (or a subscriber) persistently activate an existing service or change a subscription to activate an existing service when a subscriber logs in, the SAE determines whether the service is specified in one or more mutex groups.
2. The SAE determines how each mutex group that lists the service is configured, and the SRC software acts accordingly.
 - If all the mutex groups that list the service allow automatic deactivation of services, the SRC software removes the persistent activations for the service and changes activate-on-login subscriptions to manual.
 - If any of the mutex groups does not allow automatic deactivation of services, the SRC software will not allow you to:
 - Persistently activate the service.
 - Change the subscription to activate the service when a subscriber logs in.

- Related Topics**
- Overview of Restricting and Customizing Services for Subscribers on page 28
 - Overview of Restricting Simultaneous Activation of Services on page 26
 - Restricting Service Activation on page 33
 - Adding a Mutex Group (SRC CLI) on page 27

Adding a Mutex Group (SRC CLI)

Use the following configuration statements to configure a mutex group in the global service scope:

```
services global mutex-group name {
  auto-deactivate (yes | no);
  description description ;
  services [ services... ];
}
```

Use the following configuration statements to configure a mutex group in a service scope:

```
services scope name mutex-group name {
  auto-deactivate (yes | no);
  description description ;
  services [ services... ];
}
```

To add a mutex group:

1. From configuration mode, enter the mutex group configuration. In this sample procedure, the mutex group is called Video.

```
user@host# edit services global mutex-group Video
```

2. Configure the method that the SAE uses to manage activation of services defined in this group.

```
[edit services global mutex-group Video]
user@host# set auto-deactivate (yes | no)
```

3. Enter a description for the service.

```
[edit services global mutex-group Video]
user@host# set description description
```

4. Configure the lists of services that the mutex group contains.

```
[edit services global mutex-group Video]
user@host# set services [ services... ]
```

5. (Optional) Verify your configuration.

```
[edit services global mutex-group Video]
user@host# show
auto-deactivate yes;
description "Video Services providing access to the same site with
different quality";
services [ Video-Bronze Video-Gold Video-Silver ];
```

- Related Topics**
- Overview of Restricting Simultaneous Activation of Services on page 26
 - Customizing Service Implementations on page 22
 - Adding a Mutex Group (C-Web Interface)
 - Restricting Simultaneous Activation of Persistent or Automatic Services on page 27

Customizing Service Delivery with Scopes

- Overview of Restricting and Customizing Services for Subscribers on page 28

Overview of Restricting and Customizing Services for Subscribers

Service scopes let you customize which services are to be delivered to specific organizations or specific locales. You can use service scopes to provision services for a group of subscribers by specifying:

- Particular services or mutex groups.
- Parameter substitutions that customize generic services.

A service scope is a collection of services and mutex groups, and optionally defines parameter substitutions for its associated services. For more information about

parameter substitutions, see “Parameters and Substitutions” on page 209. The object *o = Services* is the generic service scope—a collection of services and mutex groups available to all subscribers.

You can assign service scopes to virtual routers (VRs) and to some types of subscribers.

Assigning Service Scopes to Multiple VRs and Subscribers

You can also assign a service scope to multiple VRs and subscribers. For example, by assigning a service scope to a group of VRs, you can specify that a service is available only in the locations served by those VRs. If a subscriber of this service accesses the network from a location where you do not offer this service, the portal will not display the service, and the subscriber will not be able to use it.

If you assign a service scope to multiple VRs and subscribers, you specify a precedence—a numerical ranking—for each service scope. The lower the precedence value, the higher the ranking of the service scope. By default, the object *o = Services* has the highest precedence value and the lowest ranking.

Defining Multiple Scopes for a Service

If multiple service scopes that define the same service are assigned to a VR or subscriber, the SAE selects the parameters to use for the service as follows:

1. It selects the parameters that are defined by only one service scope.
2. If the same parameter is defined by more than one service scope, the SAE selects the parameter as follows:
 - a. Selects the parameter associated with the service scope that has the lowest precedence value.
 - b. If the parameter is defined by multiple service scopes with the same precedence value, selects the parameter defined by the service scope with the lowest alphanumerical name.

For example, consider the situation shown in Table 5 on page 29 in which three scopes define several parameters for the same service.

Table 5: Parameter Selection Example

Service Scope Name	Precedence Value	Parameter Definitions
s1	1	description, policy group
s2	5	description, URL
s3	5	description, URL

The SAE will use the following parameter definitions for the service:

- Description from scope s1 (s1 has the lowest precedence value)

- Policy group from scope s1 (only s1 defines this parameter)
- URL from scope s2 (s2 has a lower alphanumeric name than s3)

You can also configure a generic Internet access service, and use service scopes to define the access parameters for different locations to use this service. If multiple service scopes that define this Internet access service are assigned to a VR, the SAE uses the precedence values to determine how to customize the service.

Example: Using Service Scopes to Deliver a Limited Set of Services to Organizations

You can use service scopes to create a limited set of services to be made available to specified organizations. For enterprise users, you could define a set of services available on the JUNOS routing platform.

To deliver a small set of services to specified enterprises:

1. Create a scope for the services to be made available. For example, see the EntJunos scope in the sample data.

```
user@host> show configuration services scope EntJunos
```

2. Add services to the scope, such as those in the sample data in the EntJunos scope.
3. Assign the scope to one or more enterprise subscribers. For example, assign the EntJunos scope to the Acme enterprise.

```
user@host# edit subscribers retailer ENT subscriber-folder entAcme enterprise Acme
```

```
[edit subscribers retailer ENT subscriber-folder entAcme enterprise Acme]
user@host# set scope EntJunos
```

4. Verify your configuration.

```
[edit subscribers retailer ENT subscriber-folder entAcme enterprise Acme]
user@host# show
scope EntJunos;
```

If you use a portal to manage enterprises, you see only the services for the specified scope from the portal. Other services are not visible to the IT managers who manage services and subscriptions from the enterprise service portal. To see the services available to Acme from Enterprise Manager Portal, see the *SRC-PE Sample Applications Guide*.

Example: Using Service Scopes to Customize Generic Services to Particular Regions

You could use service scopes to customize a generic audio service called Audio-Bronze on a regional basis. This example assumes that the network is configured so that VR boston serves the Boston subnet and VR chicago serves the Chicago subnet.

When the network starts operating, the SAE substitutes the parameters you specified in the service scope definition for the corresponding fields in the service subordinate to that scope.

To customize the new service Audio-Bronze for the Boston and Chicago subnets:

1. Add the Audio-Bronze service within a service scope called boston, and configure the IP address and mask used by VR boston in the parameter configuration.

This IP address and mask determine an access point to the service provider's equipment.

```
user@host# edit services scope boston
```

```
[edit services scope boston]
```

```
user@host# edit service Audio-Bronze
```

```
[edit services scope boston service Audio-Bronze]
```

```
user@host# set parameter service-ip-address 10.10.40.33
```

```
[edit services scope boston service Audio-Bronze]
```

```
user@host# set parameter service-ip-mask 255.255.255.255
```

2. Add another Audio-Bronze service within a service scope called scope_chicago, and specify the IP address and mask used by VR chicago.

```
user@host# edit services scope chicago
```

```
[edit services scope chicago]
```

```
user@host# edit service Audio-Bronze
```

```
[edit services scope chicago service Audio-Bronze]
```

```
user@host# set parameter service-ip-address 10.10.55.1
```

```
[edit services scope chicago service Audio-Bronze]
```

```
user@host# set parameter service-ip-mask 255.255.255.255
```

3. Assign service scope boston to virtual router boston.

```
user@host# edit shared network device region_one virtual-router boston
```

```
[edit shared network device region_one virtual-router boston]
```

```
user@host# set scope boston
```

4. Assign service scope chicago to virtual router chicago.

```
user@host# edit shared network device region_two virtual-router chicago
```

```
[edit shared network device region_two virtual-router chicago]
```

```
user@host# set scope chicago
```

- Related Topics**
- Overview of Restricting Simultaneous Activation of Services on page 26
 - Overview of SRC Aggregate Services on page 10
 - Configuring Service Scopes (SRC CLI) on page 32
 - Restricting Service Activation on page 33

Configuring Service Scopes (SRC CLI)

1. Adding Service Scopes (SRC CLI) on page 32
2. Assigning Services and Mutex Groups to Service Scopes (SRC CLI) on page 32
3. Assigning Service Scopes to VRs or Subscribers (SRC CLI) on page 33

Adding Service Scopes (SRC CLI)

Use the following configuration statement to configure service scopes:

```
services scope name {
    precedence precedence ;
}
```

To add a service scope:

1. From configuration mode, enter the service scope configuration. In this sample procedure, the scope is called EntJunos.

```
user@host# edit services scope EntJunos
```

2. Configure the precedence of the service scope.

```
[edit services scope EntJunos]
user@host# set precedence precedence
```

3. (Optional) Verify your configuration.

```
[edit services scope EntJunos]
user@host# show
precedence 2;
```

Assigning Services and Mutex Groups to Service Scopes (SRC CLI)

To assign services and Mutex Groups to a scope:

- Add the service or mutex group at the edit services scope hierarchy level.

For example, to add a service to a service scope called video, enter the following:

```
user@host# edit services scope video service Video-Gold
```

Assigning Service Scopes to VRs or Subscribers (SRC CLI)

You can assign multiple service scopes to a VR or subscriber, and you can assign a service scope to multiple VRs and subscribers.

To assign a service scope:

1. Enter the configuration for the object to which you want to add the service scope. For example:

```
user@host# edit shared network device erx-node1 virtual-router default
```

2. Assign a scope to the object.

```
[edit shared network device erx-node1 virtual-router default]
user@host# set scope scope
```

- Related Topics**
- Overview of Restricting and Customizing Services for Subscribers on page 28
 - Customizing Service Implementations on page 22
 - Restricting Service Activation on page 33
 - Configuring Service Scopes (C-Web Interface)

Restricting Service Activation

You can configure services that cannot be deactivated by an SAE API call; for example, service deactivated from a portal application. This feature is useful when a subscriber has access to several services that perform similar functions, and must use one and only one of those services at a time.

In this case, you must complete three actions:

1. Configure one of the services as a permanent service. This configuration causes the SAE to activate one of the services automatically when the SAE creates a subscriber session.
2. Configure each service to be activate only. This configuration prevents the SAE from deactivating the only active service of this type.
3. Add all services to a mutex group. This configuration allows the SAE to activate one of the other services and to deactivate the service that is currently active.

For example, a subscriber may be able to use one of three Internet access services, each of which offers different speeds. If you configure one of these services as a permanent service, the SAE activates this service for the subscriber automatically. Because all Internet access services are marked to be activate only, the subscriber cannot request deactivation of the default Internet access service. However, if the subscriber requests a faster Internet access service, the SAE activates the faster service and deactivates the default service, because the SAE cannot allow concurrent activation of multiple services assigned to the same mutex group.

- Related Topics**
- Restricting Simultaneous Activation of Persistent or Automatic Services on page 27
 - Overview of Restricting Simultaneous Activation of Services on page 26
 - Overview of Restricting and Customizing Services for Subscribers on page 28

Chapter 2

Managing Service Schedules

- Overview of Service Schedules on page 35
- Schedule Configuration Guidelines on page 40
- Planning Service Schedules on page 41

Overview of Service Schedules

Service schedules define when specified services will be activated or deactivated and can also indicate when specified services are available or unavailable to subscribers. You can configure a service schedule for all subscribers to a service, or for a selected subscriber or subscribers. Schedules are composed of a number of rules expressed as schedule entries in schedule configuration.

You can exclude specified times, such as a day of the week, a specific date, or a time interval, from schedule rules. These times are referred to as schedule exclusions.

There are three types of schedules:

- Event-based schedules—The SAE activates or deactivates a service at a specified time. You specify the time the action is to occur, and any intervals to extend that time.
- Authorization schedules—The SAE allows or disallows access to a service during a specified interval; it can also deactivate sessions for current subscribers to a service at the beginning or end of an interval.
- State-based schedules—The SAE controls the times at which a service is available. Subscribers cannot change these schedules.

Event-Based Schedules

For each rule in event-based schedules, you specify a time at which the SAE activates or deactivates a specified service. In most cases for schedules configured under the global service configuration (for example, *o = Services*), a subscriber must be logged in at the time that the event occurs. For example, if a service is scheduled to be activated at 8 AM, the subscriber must already be logged in to the system at 8 AM.

You can extend the time at which a scheduled action can be initiated by configuring the following for event-based schedules:

- **Action threshold**—Interval after a scheduled time that an action can occur. The action threshold is configured globally for the SAE server.
- **Preparation time**—Interval before a scheduled time that an action can occur. The preparation time is configured globally for the SAE server.

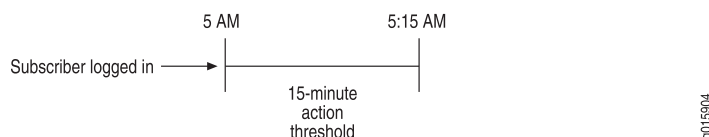
Extending the time gives subscribers flexibility in when they can log in and in the time they can perform a task. It also gives the system time to complete a transition from one state to another and distributes the load on the system.

You can also configure an interval after a scheduled time that an action can occur for individual schedules and event-based schedules.

Action Threshold

The action threshold indicates the maximum delay that a service allows for a time-related change to occur. For example, you can allow a 15-minute delay so that if an event is scheduled for 5:00 AM but the system is not able to perform the event at 5:00 AM, the SAE attempts to perform the action until 5:15 AM, as shown in Figure 3 on page 36.

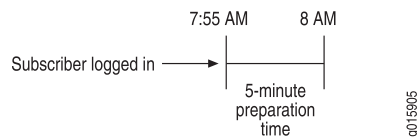
Figure 3: Sample Action Threshold



Preparation Time

Because the transition from one state to another does not occur instantaneously, the SAE uses the preparation time to allow for the time that the SAE needs to make the transition. For example, if you have a pay-per-view service and many subscribers need to have the service activated by a certain time, you can configure the service schedule preparation time to begin the process early to make sure that everyone gets his or her service activated by the time the event starts. Or you could schedule a few minutes of preparation time for setting up a videoconference.

A preparation time applies only to subscribers who have a service schedule and who are logged in to their subscriber session before the preparation time starts. For example, if you define a service schedule that activates service Audio-Gold at 8:00 AM, this service is activated only for subscribers who are subscribed to this service and are logged in as of 7:55 AM (assuming a default preparation time of 5 minutes). The service is not activated for subscribers who log in between 7:55 AM and 8:00 AM, as shown in Figure 4 on page 37.

Figure 4: Sample Preparation Time

Authorization Schedules

For authorization schedules, a service is either available or unavailable. You can configure intervals during which subscribers can log in and activate a specified service and intervals during which subscribers cannot activate a specified service. In addition, an authorization schedule can deactivate a service at a specified time for subscribers who are using the service.

For example, you could use an authorization schedule to offer a service only between 5 PM and 8 PM. In this case, you can configure a schedule that denies activation of the service during any other time period. If a subscriber attempts to activate the service at a time other than between 5 PM and 8 PM, the activation is denied.

You can configure authorization schedules only for services that use authorization; that is, a service configured to use an authorization plug-in, such as the `scheduleAuth` plug-in provided by the sample data.

State-Based Schedules

For state-based schedules, you create service schedules that are controlled administratively. A state-based schedule defines when a service is available or unavailable.

For example, you could configure a schedule to provide a service at 5 Mbps from 8 AM to 4 PM and another service at 2 Mbps from 3:45 PM to 8:15 AM. The time overlap ensures that one of the services is available at transition time.

You create state-based service schedules from:

- Enterprise Manager Portal—Service providers make schedules available to IT managers in enterprises. IT managers can then configure service schedules for their enterprises.

See the *SRC-PE Sample Applications Guide*.

- An application that uses the CORBA remote API—You can incorporate service schedules, including schedules that affect subscriber sessions, in an application that has been created with the CORBA remote API, such as a residential portal.



NOTE: The only way to associate a session with a service schedule is through the CORBA remote API.

For information about the residential portal, see *SRC-PE Sample Applications Guide*.

For information about the SAE CORBA remote API, see the documentation for the API on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>.

Effective Period for Service Activation or Deactivation

You can configure an effective period for a schedule rule to give subscribers an opportunity to take advantage of a scheduled action for a specified amount of time, rather than for one specific time. If users log in after a scheduled action but before the end of the effective period, they can take advantage of the service. Although similar to an action threshold, an effective period can be configured for each schedule rule, whereas the action threshold applies to all schedules on an SAE.

An effective period is active for service schedules assigned to subscribers under the subscriber tree (for example, *o = Users*), but not for services under the global service configuration or a defined service scope (for example, *o = Services* or *o = Scopes*).

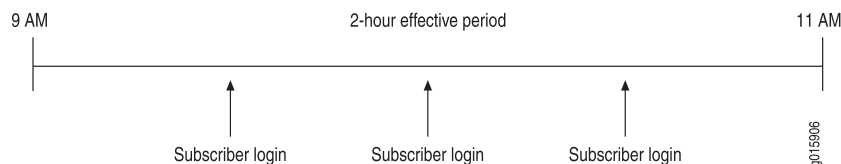
An effective period applies to subscribers who:

- Have a service schedule that includes an effective period
- Are logging in to their subscriber session

An effective period does not apply to subscribers who are already logged in to the system.

For example, you could create a schedule that includes a scheduled event to start at 9 AM and an effective period of 2 hours; subscribers can log in between 9 AM and 11 AM and have the event take place, as shown in Figure 5 on page 38.

Figure 5: Sample Effective Period



You can use effective periods rather than activate-on-login for subscriptions. If activate-on-login is configured for a subscription, we recommend that the service for the subscription not have an effective period configured.



NOTE: If an effective period is configured so that it overlaps with an excluded time, the scheduled event does not take place, because it is within an excluded time period. To clearly define when a scheduled event can occur, do not configure an effective period to overlap with an excluded time.

One-Time Events and Recurring Events

You can specify service schedules for numerous situations. For example, you can set up:

- A one-time event—Performs an action at a specified time; for example, activating a gold Internet service at 7:00 AM on January 1, 2006.
- A recurring event—Performs an action over a period of time at specified intervals; for example, activating a gold video service at 7:00 AM every morning.
- A working-hours service—Performs actions at specified times on Monday through Friday; for example, a gold Internet service that is activated Monday through Friday at 8:00 AM and deactivated Monday through Friday at 5:00 PM. This type of service requires two schedule entries—one that activates the service and one that deactivates the service.

Schedule Availability to Subscribers

Which subscribers a service schedule affects depends on the configuration for the schedule. Table 6 on page 39 shows which subscribers are affected by a schedule.

Table 6: Schedule Availability to Service Subscribers

Schedule Configured for This Object	Applies to These Subscribers
Service	All subscribers to that service
Scope	All subscribers to the specified service in that scope
Retailer	Any subscriber subordinate to the retailer for whom the service schedule is configured
Subscriber	The subscriber for whom the service schedule is configured or, in the case of enterprise subscribers, any subscribers subordinate to that subscriber

When a service provider or IT manager creates a schedule and attaches it to a service, the service schedule can be assigned to enterprise subscribers or residential subscribers. In some instances, subscribers can also create their own service schedules. When the scheduled action occurs, it applies to subscribers who are logged in and have a subscription to the scheduled service.

Schedule Exclusions

You can also exclude specific time intervals from a service schedule. For example, you can set:

- A holiday schedule—Ignores the service schedule for a specified day; for example, for January 1.

- A promotional period—Ignores the service schedule for a specified interval; for example, a 2-week period after the start date for the promotion.

Excluded times can apply to event schedules and authorization schedules. You can create numerous exclusion intervals to specify different actions and different times.

- Related Topics**
- Schedule Configuration Guidelines on page 40
 - Planning Service Schedules on page 41
 - Setting the Action Threshold and Preparation Time (SRC CLI) on page 43
 - Authorizing Scheduled Services (SRC CLI) on page 44
 - Adding a Service Schedule (SRC CLI) on page 45

Schedule Configuration Guidelines

Use the following guidelines when you plan and configure service schedules:

- Do not configure schedules for services that are configured as persistent services on the router.
- If activate-on-login is configured for a subscription, do not configure an effective period in a schedule for the associated service.

Consider changing the configuration for this subscription to use an effective period, rather than activate-on-login.

- Make sure you know the values for preparation time and action threshold that have been configured for the SAE.
- Do not configure an effective period to overlap with an excluded time.
- To avoid schedule conflicts, configure one service schedule to include all rules that control a service.
- Determine whether or not a service to be scheduled has an authorization plug-in configured. If an authorization plug-in is configured for a service, you can create an authorization schedule for that service.
- Create a schedule for a service under one of the following:
 - The subscriber tree (for example, *o = Users*)
 - The global service configuration (for example, *o = Services*)
 - A defined service scope (for example, *o = Scopes*)
- Do not specify the time in a schedule entry that is more than 5 years in the past or 15 years in the future.

- Related Topics**
- Overview of Service Schedules on page 35
 - Planning Service Schedules on page 41
 - Authorizing Scheduled Services (SRC CLI) on page 44

Planning Service Schedules

Before you configure service schedules, carefully plan individual rules for the schedule to avoid conflicts between the rules. The rules become entries when you configure the schedule. The SAE evaluates each schedule entry independently of the others.

The following list of planning activities applies to both event-based and authorization schedules unless otherwise indicated.

For each service schedule:

1. Decide whether to configure the schedule for a group of subscribers. Configure a schedule that includes rules for the same service under only one of the following:
 - The global service configuration (for example, *o = Services*)
 - A defined service scope *o = Scopes*
 - The subscriber tree *o = Users*
2. For each rule in a service schedule, list the following information for each service included in the schedule:
 - Time to activate the service and any effective time associated with this action.
 - Time to deactivate the service.

or

(Optional for authorization schedules) Time to deny or to deny and deactivate the service.

Times can include a date and day of the week.

3. (Event-based schedules) Make sure that the scheduled times take into consideration a preparation time or an action threshold that has been configured for the SAE.

For example, if a schedule entry activates a service at 8:00, a schedule entry to deny access to the service should have a time before 8:00, such as 7:59. If a preparation time of 15 minutes is configured for the SAE, a schedule entry to deny access to the service should have a time before 7:45. The deny period ends before the service can be activated, with the time between the end of the deny interval and the activation time greater than the preparation time.

4. List any exclusions to a schedule, including:
 - Time the exclusion starts
 - Time the exclusion ends

Times can include a date and day of the week.

5. Review all rules for the schedule, and make sure that individual rules do not conflict with one another. Make sure that activate and deactivate times do not overlap for the same service.

- Related Topics**
- Overview of Service Schedules on page 35
 - Schedule Configuration Guidelines on page 40
 - Adding a Service Schedule (SRC CLI) on page 45
 - Authorizing Scheduled Services (SRC CLI) on page 44

Chapter 3

Scheduling Services (SRC CLI)

- Setting the Action Threshold and Preparation Time (SRC CLI) on page 43
- Authorizing Scheduled Services (SRC CLI) on page 44
- Adding a Service Schedule (SRC CLI) on page 45
- Setting the Time Schedule (SRC CLI) on page 47
- Guidelines for Entering Time Values for Service Schedules on page 49
- Setting the Action for a Service Schedule (SRC CLI) on page 50
- Defining Attributes for Service Activation (SRC CLI) on page 51
- Example: Configuring Different Service Tiers for Different Days with the CLI on page 51
- Example: Configuring a Service to Be Active During Nonwork Hours with the CLI on page 53
- Example: Configuring a Service to Be Available for a Specified Interval with the CLI on page 55

Setting the Action Threshold and Preparation Time (SRC CLI)

You can set the action threshold and preparation time for all schedules; you cannot set these values for individual schedules.

Use the following configuration statements to set the action threshold and preparation time:

```
shared sae configuration time-based-policies {  
    action-threshold action-threshold;  
    preparation-time preparation-time;  
    max-worker-threads max-worker-threads;  
}
```

To set the action threshold and preparation time for an SAE:

1. From configuration mode, access the configuration statement that configures time-based policies.

```
user@host# edit shared sae configuration time-based-policies
```

2. Configure the maximum delay that the service allows for a time-related change to occur. The recommended range is 60000–300000 milliseconds. The minimum value supported is 60000 milliseconds.

```
[edit shared sae configuration time-based-policies]
user@host# set action-threshold action-threshold
```

3. Configure the preparation time permitted for a state transition.

```
[edit shared sae configuration time-based-policies]
user@host# set preparation-time preparation-time
```

When you set a value for the preparation time, take into consideration system load and performance. Factors such as the number of subscribers, the number of active services, the number of schedule services, the speed of the processor on the system, as well as other conditions might affect the amount of time to process all the scheduled actions at a specified schedule time.

4. (Optional) Configure the maximum number of threads for service scheduling.

```
[edit shared sae configuration time-based-policies]
user@host# set max-worker-threads max-worker-threads
```

5. (Optional) Verify your configuration.

```
[edit shared sae configuration time-based-policies]
user@host# show
```

- Related Topics**
- Overview of Service Schedules on page 35
 - Adding a Service Schedule (SRC CLI) on page 45
 - Setting the Action Threshold and Preparation Time with the C-Web Interface
 - Setting the Time Schedule (SRC CLI) on page 47
 - Setting the Action for a Service Schedule (SRC CLI) on page 50

Authorizing Scheduled Services (SRC CLI)

You can configure an authorization plug-in to authorize a scheduled service by specifying the name of the plug-in that authorizes the schedule in the service definition. The default schedule authorization plug-in is named `scheduleAuth`.

Use the following configuration statement to configure an authorization plug-in for a service configured in the global configuration:

```
services global service name {
  authorization-plug-in [authorization-plug-in...];
}
```

Use the following configuration statement to configure an authorization plug-in for a service configured in the service scope:

```
services scope name service name {
  authorization-plug-in [authorization-plug-in...];
}
```

To define an authorization plug-in for a service:

1. From configuration mode, access the configuration statement that configures the service configuration in the global configuration or in the service scope.

```
user@host# edit services global service name
user@host# edit services scope name service name
```

For example, to configure the service named Video-Gold in the global configuration:

```
user@host# edit services global service Video-Gold
```

2. Enter the name of the authorization plug-in that will authorize the schedule for this service.

```
user@host# set authorization-plug-in [authorization-plug-in...]
```

For example, to specify the default schedule authorization plug-in:

```
user@host# set authorization-plug-in scheduleAuth
```

- Related Topics**
- Overview of Service Schedules on page 35
 - Adding a Service Schedule (SRC CLI) on page 45
 - Authorizing Scheduled Services (C-Web Interface)
 - Defining Attributes for Service Activation (SRC CLI) on page 51

Adding a Service Schedule (SRC CLI)

You can create a service schedule for the following objects:

- Scopes
- Services
- Retailers
- Enterprises
- Subscribers in an enterprise



NOTE: If you change or remove the name of a service that is referenced by a schedule, the SRC software treats this case like one in which no subscribers have a subscription to this service. In both cases, the action for the service is not taken. The software does not regard either case as an error in the schedule; a failure is not reported.

Use the following statements to configure a service schedule:

```
schedule name {
    description description;
}
```

To add a service schedule:

1. From configuration mode, access the configuration statement that configures the service schedule for the objects for which you can create a service schedule. Enter a unique name for the service schedule.

For example:

```
user@host# edit services scope name schedule name
```

```
user@host# edit services global schedule name
```

```
user@host# edit subscribers retailer name schedule name
```

```
user@host# edit subscribers retailer name subscriber-folder folder-name
enterprise name schedule name
```

```
user@host# edit subscribers retailer name subscriber-folder folder-name
subscriber name schedule name
```

2. (Optional) Describe the service schedule.

```
user@host# set description description
```

3. Create schedule entries for the service schedule. A number of schedule entries, or rules, constitute each service schedule.

```
user@host# set event name
```

An entry consists of the schedule time, any excluded times, and a list of actions. To create an entry:

- Specify the time schedule.

See “Setting the Time Schedule (SRC CLI)” on page 47.

- Specify the actions.

See “Setting the Action for a Service Schedule (SRC CLI)” on page 50.

Related Topics

- Overview of Service Schedules on page 35
- Planning Service Schedules on page 41
- Authorizing Scheduled Services (SRC CLI) on page 44
- Adding a Service Schedule (C-Web Interface)

- Example: Configuring a Service to Be Active During Nonwork Hours with the CLI on page 53
- Example: Configuring a Service to Be Available for a Specified Interval with the CLI on page 55

Setting the Time Schedule (SRC CLI)

When you set up a time schedule, you specify:

- For event schedules—Time at which an action is to occur; the from date and time information
- For schedules for services that have authorization configured—Beginning and end of the interval; the to date and time information
- For exclusions—Times to be excluded from that schedule

Use the following statements to configure a time schedule for an event:

```

schedule name event name from {
    effective effective;
    hour hour;
    minute minute;
    day-of-month day-of-month;
    day-of-week day-of-week;
    month month;
    year year;
    time-zone time-zone;
}
schedule name event name to {
    effective effective;
    hour hour;
    minute minute;
    day-of-month day-of-month;
    day-of-week day-of-week;
    month month;
    year year;
    time-zone time-zone;
}

```

Use the following statements to configure time exclusions from the schedule:

```

schedule name event name except name from {
    hour hour;
    minute minute;
    day-of-month day-of-month;
    day-of-week day-of-week;
    month month;
    year year;
    time-zone time-zone;
}
schedule name event name except name to {
    hour hour;
}

```

```

minute minute;
day-of-month day-of-month;
day-of-week day-of-week;
month month;
year year;
time-zone time-zone;
}

```

To configure the time schedule:

1. From configuration mode, access the configuration statement that configures the service schedule for the objects for which you can create a service schedule. Enter a name for the event and the exclusion. The specified name is not stored as an identifier, so the arbitrary value can be as simple as a number.
2. (Optional) Specify the effective period in which to schedule the event. This period is the interval after the associated from or to time during which the scheduled action can be initiated by a subscriber who is logging in to a subscriber session.

```
user@host# set effective effective
```

3. (Optional) Specify the hour of the day in the indicated month in which to schedule the event or exclusion.

```
user@host# set hour hour
```

4. (Optional) Specify the minutes past the indicated hour in which to schedule the event or exclusion.

```
user@host# set minute minute
```

5. (Optional) Specify the day of the month in which to schedule the event or exclusion.

```
user@host# set day-of-month day-of-month
```

6. (Optional) Specify the day of the week in which to schedule the event or exclusion.

```
user@host# set day-of-week day-of-week
```

7. (Optional) Specify the month of the year in which to schedule the event or exclusion.

```
user@host# set month month
```

8. (Optional) Specify the year in which to schedule the event or exclusion.

```
user@host# set year year
```

9. (Optional) Specify the time zone to use in the schedule.

```
user@host# set time-zone time-zone
```

- Related Topics**
- Overview of Service Schedules on page 35
 - Guidelines for Entering Time Values for Service Schedules on page 49
 - Adding a Service Schedule (SRC CLI) on page 45
 - Setting the Action for a Service Schedule (SRC CLI) on page 50

Guidelines for Entering Time Values for Service Schedules

When you enter time schedules, you can use the values in the following list. See “Setting the Time Schedule (SRC CLI)” on page 47 for a description of the options.



NOTE: Dates in the **to** statements apply only to services that have an authorization plug-in configured. If an authorization plug-in is not configured for the service associated with the schedule, the entries in the **to** statements are ignored.

- *—Asterisks are interpreted as follows:
 - Minutes and hours:
 - 0 if used in the **from** or **to** statements of a scheduled event
 - First or last if used in the statements of a schedule exclusion
 - Time zones—Local SAE time zone
 - All other options—First through last
 - For options in the **to** statements, * for the end time is equivalent to “deny service activation after this start date.”
 - For dates in the **from** statements, * is equivalent to “deny service activation before this end date.”
 - Range of numbers separated by a hyphen. The range is inclusive; for example, 1-5 for the hour specifies hours 1, 2, 3, 4, and 5.
 - List of numbers or ranges separated by commas. For example, 1,2,5,9 or 0-4,8-12.
 - Skip values in ranges:
 - To skip a number’s value through the range, follow a range with / < number > . For example, 0-23/2 used in the **hour** option specifies that the event occurs every other hour.
 - Skip values with *. If you want to specify every two hours, use */2.
-



NOTE: If you set both a day of the month and a day of the week, the day of the month is used.

- Related Topics**
- Overview of Service Schedules on page 35
 - Adding a Service Schedule (SRC CLI) on page 45
 - Schedule Configuration Guidelines on page 40

Setting the Action for a Service Schedule (SRC CLI)

Use the following configuration statements to configure the list of actions for the service schedule:

```
schedule name event name action name {
    type (activate | deactivate | deny | deny-deactivate);
    service service;
}
```

To configure the actions:

1. From configuration mode, access the configuration statement that configures the service schedule for the objects for which you can create a service schedule. Enter a name for the event and the action. The specified name is not stored as an identifier, so the arbitrary value can be as simple as a number.
2. Specify the type of action. The deny and the deny-deactivate values apply only to services that have an authorization plug-in configured. For more information, see “Authorizing Scheduled Services (SRC CLI)” on page 44 .

```
user@host# set type (activate | deactivate | deny | deny-deactivate)
```

3. Specify the name of the service.

```
user@host# set service service
```

4. (Optional) Specify substitutions to be used when the service is activated. Substitutions apply only to service activations.

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

For more information, see the activateService method of the SAE external interface in the SAE CORBA remote API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>.

For more information about substitutions and schedules, see “Example: Configuring Different Service Tiers for Different Days with the CLI” on page 51 .

For information about the syntax for substitutions, see “Parameters and Substitutions” on page 209.

- Related Topics**
- Overview of Service Schedules on page 35
 - Adding a Service Schedule (SRC CLI) on page 45

- Setting the Time Schedule (SRC CLI) on page 47
- Setting the Action (C-Web Interface)

Defining Attributes for Service Activation (SRC CLI)

Use the following statement to configure attributes for service activation:

```
schedule name event name action name attribute (sessionName | sessionTag |
  sessionTimeout | downStreamBandwidth | upStreamBandwidth) {
  value;
}
```

To define the attributes:

1. From configuration mode, access the configuration statement that configures the service schedule for the objects for which you can create a service schedule.
2. Specify the value for the attribute that is set before the service is activated.

```
user@host# set attribute (sessionName | sessionTag | sessionTimeout |
  downStreamBandwidth | upStreamBandwidth) value
```

Subscription attributes apply only to service activations.

- Related Topics**
- Adding a Service Schedule (SRC CLI) on page 45
 - Overview of Service Schedules on page 35
 - For more information about subscription attributes, see the *Subscription.html* file in the SAE core portal API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>

Example: Configuring Different Service Tiers for Different Days with the CLI

This example shows how to configure a schedule for an audio service to provide:

- Gold level of service on weekends
- Bronze level of service on weekdays

The sample schedule:

- Uses the Audio-Gold and Audio-Bronze services in the sample data.
- Activates the Audio-Gold service and denies the Audio-Bronze service on Saturday.
- Activates the Audio-Bronze service and denies and deactivates the Audio-Gold service on Monday.
- Does not have a preparation time configured for the SAE.

For demonstration purposes, the sample schedule is configured in the global configuration to make the service schedule available to all subscribers to the two

audio services. It is assumed that subscribers are continuously logged in to the system to access the audio services.

To configure a schedule to make the Audio-Gold service available on Saturday and Sunday and the Audio-Bronze service available for the rest of the week:

1. From configuration mode, access the configuration statement that configures the service schedule in the global configuration. Enter a unique name for the service schedule; for example, `audioSchedule`.

```
user@host# edit services global schedule audioSchedule
```

Enter a description of the schedule.

```
[edit services global schedule audioSchedule]
user@host# set description description
```

2. From configuration mode, access the configuration statement that configures the schedule entry. Enter a name for the schedule entry; for example, `audioTime1`.

```
user@host# edit services global schedule audioSchedule event audioTime1
```

3. For the time, specify the day of the week as Saturday. For the actions, specify **activate** for the Audio-Gold service (named Action-1) and **deny-deactivate** for the Audio-Bronze service (named Action-2).

```
[edit services global schedule audioSchedule event audioTime1]
user@host# set from day-of-week 6
user@host# set action action-1 type activate service Audio-Gold
user@host# set action action-2 type deny-deactivate service Audio-Bronze
```

4. From configuration mode, access the configuration statement that configures the schedule entry. Enter a name for the schedule entry; for example, `audioTime2`.

```
user@host# edit services global schedule audioSchedule event audioTime2
```

5. For the time, specify the day of the week as Monday. For the actions, specify **activate** for the Audio-Bronze service (named Action-1) and **deny-deactivate** for the Audio-Gold service (named Action-2).

```
[edit services global schedule audioSchedule event audioTime2]
user@host# set from day-of-week 1
user@host# set action action-1 type activate service Audio-Bronze
user@host# set action action-2 type deny-deactivate service Audio-Gold
```

- Related Topics**
- Adding a Service Schedule (SRC CLI) on page 45
 - Example: Configuring a Service to Be Active During Nonwork Hours with the CLI on page 53
 - Example: Configuring a Service to Be Available for a Specified Interval with the CLI on page 55

Example: Configuring a Service to Be Active During Nonwork Hours with the CLI

This example shows how to configure a schedule for an Internet gold service to be active:

- Monday–Friday outside the 8:30 AM to 4:30 PM work day
- January 1 of the following year—All day

The example uses the Internet-GoldAuth service. This service is based on the Internet-Gold service in the sample data with the addition of the scheduleAuth plug-in defined as the authorization plug-in for the service.

The sample schedule:

- Deactivates the Internet-GoldAuth service from 8:30 AM through 4:29 PM.
- Activates the service at 4:30 PM.
- Does not have a preparation time configured for the SAE.

This configuration avoids schedule overlap.

For demonstration purposes, the sample schedule is configured in the global configuration to make the service schedule available to all subscribers to the Internet-GoldAuth service.

To configure a schedule to make a service available outside work hours and on January 1:

1. From configuration mode, access the configuration statement that configures the service configuration named Internet-GoldAuth in the global configuration. Specify the default schedule authorization plug-in.

```
user@host# edit services global service Internet-GoldAuth
```

```
[edit services global service Internet-GoldAuth]
```

```
user@host# set authorization-plug-in scheduleAuth
```

2. From configuration mode, access the configuration statement that configures the service schedule. Enter a unique name for the service schedule; for example, afterHours.

```
user@host# edit services global schedule afterHours
```

Enter a description for the schedule.

```
[edit services global schedule afterHours]
```

```
user@host# set description description
```

3. From configuration mode, access the configuration statement that configures the schedule entry. Enter a name for the schedule entry; for example, goldTime.

```
user@host# edit services global schedule afterHours event goldTime
```

4. From configuration mode, access the configuration statement that configures the time schedule. For the time, specify the day of the week as Monday through Friday, and specify that the schedule start at 8:30 AM and end at 4:29 PM (16:29) each day.

```
user@host# edit services global schedule afterHours event goldTime from
```

```
[edit services global schedule afterHours event goldTime from]
```

```
user@host# set day-of-week 1
```

```
user@host# set hour 8
```

```
user@host# set minute 30
```

```
user@host# edit services global schedule afterHours event goldTime to
```

```
[edit services global schedule afterHours event goldTime to]
```

```
user@host# set day-of-week 5
```

```
user@host# set hour 16
```

```
user@host# set minute 29
```

5. From configuration mode, access the configuration statement that configures the exclusion. Enter a name for the exclusion; for example, exclude-1. Specify a one-time exclusion for January 1.

```
user@host# edit services global schedule afterHours event goldTime except  
exclude-1 from
```

```
[edit services global schedule afterHours event goldTime except exclude-1 from]
```

```
user@host# set month 1
```

```
user@host# set day-of-month 1
```

By excluding January 1 from the schedule, the Internet-GoldAuth service is active all day.

6. From configuration mode, access the configuration statement that configures the action. Enter a name for the action; for example, action-1. Specify **deny-deactivate** for the Internet-GoldAuth service.

```
user@host# edit services global schedule afterHours event goldTime action  
action-1
```

```
[edit services global schedule afterHours event goldTime action action-1]
```

```
user@host# set type deny-deactivate
```

```
user@host# set service Internet-GoldAuth
```

7. From configuration mode, access the configuration statement that configures the schedule entry. Enter a name for the schedule entry; for example, goldTime2.

```
user@host# edit services global schedule afterHours event goldTime2
```

8. From configuration mode, access the configuration statement that configures the time schedule. Specify 4:30 PM (that is, 16:30).

```
user@host# edit services global schedule afterHours event goldTime2 from
```



```
[edit services global schedule afterHours event goldTime2 from]
user@host# set hour 16
user@host# set minute 30
```

9. From configuration mode, access the configuration statement that configures the exclusion. Enter a name for the exclusion; for example, exclude-2. Specify a one-time exclusion for January 1.

```
user@host# edit services global schedule afterHours event goldTime2 except  
exclude-2 from
```

```
[edit services global schedule afterHours event goldTime2 except exclude-2 from]
user@host# set month 1
user@host# set day-of-month 1
```

By excluding January 1 from the schedule, the Internet-GoldAuth service is active all day.

10. From configuration mode, access the configuration statement that configures the action. Enter a name for the action; for example, action-2. Specify **activate** for the Internet-GoldAuth service.

```
user@host# edit services global schedule afterHours event goldTime2 action  
action-2
```

```
[edit services global schedule afterHours event goldTime2 action action-2]
user@host# set type activate
user@host# set service Internet-GoldAuth
```

- Related Topics**
- Adding a Service Schedule (SRC CLI) on page 45
 - Example: Configuring Different Service Tiers for Different Days with the CLI on page 51
 - Example: Configuring a Service to Be Available for a Specified Interval with the CLI on page 55

Example: Configuring a Service to Be Available for a Specified Interval with the CLI

You can use an effective period for a schedule to make a service available to subscribers who log in during a specified time period. The following example shows how to configure a schedule to make a service available from 8 AM until 4 PM.

To make a specified service available from 8 AM until 4 PM:

1. From configuration mode, access the configuration statement that configures the service schedule in the global configuration. Enter a unique name for the service schedule; for example, effectiveHours.

```
user@host# edit services global schedule effectiveHours
```

Enter a description for the schedule.

```
[edit services global schedule effectiveHours]
user@host# set description description
```

2. From configuration mode, access the configuration statement that configures the schedule entry. Enter a name for the schedule entry; for example, `availableTime`.

```
user@host# edit services global schedule effectiveHours event availableTime
```

3. From configuration mode, access the configuration statement that configures the time schedule. Specify the time when the service is first available—8 AM—and for how long the service is to be available—480 minutes.

```
user@host# edit services global schedule effectiveHours event availableTime from
```

```
[edit services global schedule effectiveHours event availableTime from]
user@host# set hour 8
user@host# set effective 480
```

4. From configuration mode, access the configuration statement that configures the action. Enter a name for the action; for example, `action-1`. Specify **activate** for the service; for example, `Internet-GoldAuth` service.

```
user@host# edit services global schedule effectiveHours event availableTime action action-1
```

```
[edit services global schedule effectiveHours event availableTime action action-1]
user@host# set type activate
user@host# set service Internet-GoldAuth
```

- Related Topics**
- Adding a Service Schedule (SRC CLI) on page 45
 - Example: Configuring Different Service Tiers for Different Days with the CLI on page 51
 - Example: Configuring a Service to Be Active During Nonwork Hours with the CLI on page 53

Part 2

Defining Policies to Manage Traffic

- Policy Management Overview on page 59
- Overview of Using Local and Global Parameters on page 81
- Configuring Local and Global Parameters (SRC CLI) on page 101
- Configuring and Managing Policies (SRC CLI) on page 105
- Policy Examples (SRC CLI) on page 195

Chapter 4

Policy Management Overview

- Policy Management Overview on page 59
- Policy Components on page 60
- Policy Information Model on page 62
- Dynamic Bandwidth Allocation Among Traffic Flows in SRC Policy Lists for JUNOS Routing Platforms on page 71
- Delivering QoS Services in a Cable Environment on page 73

Policy Management Overview

Policy management enables network service providers to configure services that customize the treatment of individual packet flows received on a subscriber's interface. The main tool for implementing policy management is a policy list. A policy list is a set of rules, each of which specifies a policy action. A rule is a policy action optionally combined with a classification.

Packets are sorted at ingress or egress into packet flows based on attributes defined in classifier control lists (CLACLs). You can apply policy lists to packets arriving and leaving an interface. You can use policy management on ATM, Frame Relay, generic routing encapsulation (GRE), IP, IPv6, Layer 2 Tunneling Protocol (L2TP), Multiprotocol Label Switching (MPLS), and virtual local area network (VLAN) traffic.

Policy management provides:

- Policy routing—Predefines a classified packet flow to a destination port or IP address. The router does not perform a routing table lookup on the packet. This provides superior performance for real-time applications.
- Bandwidth management—Rate-limits a classified packet flow at ingress to enforce ingress data rates below the physical line rate of a port. A rate-limit profile with a policy rate-limit profile rule provides this capability. You can construct policies to provide rate limiting for individual packet flows or for the aggregate of multiple packet flows. E-series router rate limits are calculated based on the layer 2 packet size.

To configure rate limiting, you first create a rate-limit profile, which is a set of bandwidth attributes and associated actions. You next create a policy list with a rule that has rate limit as the action and associate a rate-limit profile with this rule.

You can configure rate-limit profiles to provide:

- A variety of services, including tiered bandwidth service where traffic conforming to configured bandwidth levels is treated differently than traffic that exceeds the configured values and hard-limit service where a fixed bandwidth limit is applied to a traffic flow
- A TCP-friendly rate-limiting service that works in conjunction with TCP's native flow-control functionality
- (JUNOS routers) Dynamic bandwidth sharing between lower priority traffic and unused preferred bandwidth through rate limit hierarchies
- Security—Provides a level of network security by using policy rules that selectively forward or filter packet flows. You can use a filter rule to stop a denial-of-service attack. You can use secure policies to mirror packets and send them to an analyzer.
- RADIUS policy support—Enables you to create and attach a policy to an interface through RADIUS.
- Packet tagging—Enables the traffic-class rule in policies to tag a packet flow so that the Quality of Service (QoS) application can provide traffic-class queuing. Policies can perform both in-band and out-of-band packet tagging.
- Packet forwarding—Allows forwarding of packets in a packet flow.
- Packet filtering—Drops packets in a packet flow.
- Packet mirroring—Uses secure policies to mirror packets and send them to an analyzer.
- Packet logging—Logs packets in a packet flow.

Use the SRC CLI and C-Web interface to configure policies. You configure policy components, or modules, which can be combined to implement a policy. By combining the various policy components, you can deploy a wide variety of services.

Related Topics

- Policy Components on page 60
- Policy Information Model on page 62
- Delivering QoS Services in a Cable Environment on page 73
- Before You Configure SRC Policies on page 105
- Enabling the Policy Configuration on the SRC CLI on page 107

Policy Components

The policy management architecture is fully compliant with Internet Engineering Task Force (IETF) policy management standards. The SRC policy management system uses a distributed architecture with the following components:

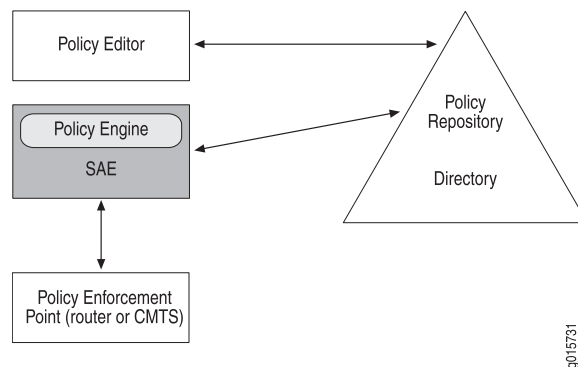
- Policy editor—Defines and deploys policies
 - Policies, Services, and Subscribers CLI

- Policies, Services, and Subscribers subtasks in the C-Web interface
- Policy engine—Resides on the SAE and makes policy decisions (policy decision point)
- Policy enforcement point—Resides on the router or policy server and performs policy management on the router
- Policy repository—Resides in the directory and stores and distributes policies

Figure 6 on page 61 shows the components of the policy management system. As shown:

1. Policy editor (Policies, Services, and Subscribers CLI or the Policies, Services, and Subscribers subtasks in the C-Web interface) is used to create policies and maintain policy data in the policy repository.
2. The policy repository distributes policy data to policy engines that are located on SAEs throughout the network.
3. The policy engine uses the policy data to instruct the policy enforcement points to apply appropriate policies to subscriber traffic in the network.

Figure 6: Policy Management Components



Policies, Services, and Subscribers CLI and the Policies, Services, and Subscribers Subtasks in the C-Web Interface

The Policies, Services, and Subscribers CLI and the Policies, Services, and Subscribers subtasks in the C-Web interface are the applications that you use to define policies. In the CLI, you can use **show** commands to view information about configured policies.

Policy Engine

The policy engine acts as a policy decision point (PDP) and is responsible for making decisions about the deployment of policies on the router or the CMTS device. The policy engine runs as part of the SAE.

Policy Repository

The policy repository is a directory that stores policies and distributes policies to policy engines.

Policy Enforcement Point

The policy enforcement point is the policy management component of the router that is responsible for enforcing the deployed policies. In cable networks, the policy enforcement point is the CMTS device.

- Related Topics**
- Enabling the Policy Configuration on the SRC CLI on page 107
 - Enabling the Policy Configuration on the C-Web Interface
 - Policy Management Overview on page 59
 - Policy Information Model on page 62
 - Before You Configure SRC Policies on page 105
 - Example: Creating Access Policies for Subscribers on page 195

Policy Information Model

Policies are made up of conditions and actions that cause the router to handle packets in a certain way.

- Condition—Defines values or fields that a packet must contain before an action is triggered; for example, packet direction, network protocol, source and destination ports, application protocol, source and destination networks, packet length, forwarding class, source and destination class
- Action—Specifies the action that the router takes on packets that match the condition; for example, filter (drop), forward, send to next interface, apply rate and burst size limits, assign a forwarding class

Here are two examples of policies with conditions and actions:

- A stateful firewall:
 - Condition—Matches input packets to a specific destination network
 - Action—Forwards matching packets
- Controlled access policy that defines the sites that a subscriber can view:
 - Condition—Traffic to and from the restricted site
 - Action—Access to the site is stopped if the site has a restricted rating

The SRC policy information model is designed to consolidate information models from various devices to provide a standard way to configure policies. This way, similar operations on different devices are represented as a single policy action or condition which is translated to device-specific operations. For example, the SRC policy information model provides an action that forwards traffic. This action is

translated into actions such as forward, accept, or simple handoff on various routers. For instances in which policy conditions or actions are significantly different, the model provides support for each type of condition or action. For example, because rate-limiting on JUNOSe routers is significantly different than policing on JUNOS routing platforms the SRC software provides a rate-limit action for JUNOSe routers and policer action for JUNOS routing platforms.

For JUNOSe routers, SRC policies are translated at the COPS-PR or COPS-XDR level, and at the router level. For JUNOS routing platforms, policies are translated at the JUNOS XML on BEEP level and at the router level.

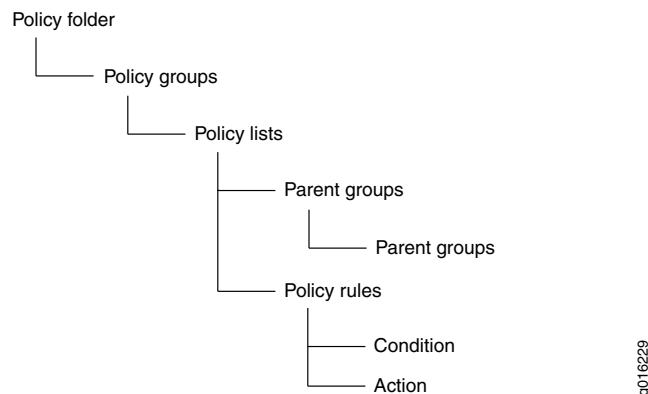
The SRC policy model also lets you simplify policy configuration for policy conditions that classify traffic. For JUNOSe and PCMM policies, you can combine different conditions that classify traffic and configure these conditions to use a single action. In addition for JUNOSe policies, you can create a condition which actually represents a number of classifiers. The SAE expands the classifier to multiple classifiers before installing them on the router. For JUNOSe routers, you can also configure hierarchical rate limits to provide dynamic bandwidth sharing.

For more information about multiple classifiers and expanded classifiers, see “Policy Components” on page 60.

Policy Objects

The SRC policy model is made up of objects that are organized as shown in Figure 7 on page 63.

Figure 7: Policy Object Organization



The following is a description of these objects:

- Policy folders—Used to organize policy groups.
- Policy groups—Hold policy lists. You associate policy groups with a service or with an interface. The SAE sends the information in a policy group to the router, and the router uses the information to create policies that it attaches to router interfaces.
- Policy lists—Used to organize policy rules. You can create policy lists for JUNOS routing platforms, for JUNOSe routers, for AAA devices, or for PCMM devices.

Whether you create a JUNOS policy list, a JUNOSe policy list, or a PCMM policy list determines the types of policy rules that you can add to the policy list.

- Parent groups—Used to develop hierarchies of rate-limit actions for policies implemented on JUNOSe routers.
- Policy rules—Used to organize the conditions and actions that make up the policy rule. Policy rules consist of conditions that you use to match traffic and actions that specify the action to take if traffic matches the condition. In JUNOS terminology, a policy rule is the same as a *term*.
- Conditions—Define match conditions or classifiers that a packet or packet flow must contain; for example, packet direction, network protocol, application protocol, source and destination networks, packet length, forwarding class, and source and destination class
- Actions—Define the action that the router or CMTS device takes on packets that match conditions

Policy Rules

JUNOSe routers support IP4 and IPv6 policy rules, PCMM devices support one type of policy rule, and JUNOS routing platforms support five types of policy rules:

- JUNOS Adaptive Services PIC (ASP)

Supports stateful firewall and Network Address Translation (NAT) services.

- JUNOS scheduler

Supports transmission scheduling and rate control parameters on interfaces that support the per-unit scheduler. Schedulers define the priority, bandwidth, delay buffer size, rate control status, and RED drop profiles to be applied to a particular class of traffic.

- JUNOS shaping

Supports setting a shaping rate on PICS that support shaping rate and on interfaces that support the per-unit scheduler.

- JUNOS filter

Supports JUNOS firewall filters.

- JUNOS policer

Supports policing, or rate limiting, by enabling you to limit the amount of traffic that passes into or out of an interface. It is an essential component of firewall filters that is designed to thwart denial-of-service attacks.

Policing applies two types of rate limits on the traffic:

- Bandwidth—Number of bps permitted, on average.
- Maximum burst size—Maximum size permitted for bursts of data that exceed the bandwidth limit.

Supported Conditions and Actions

The types of conditions and actions that are available for a policy rule depend on the type of rule. Figure 8 on page 65 shows the types of conditions and actions that are available for JUNOS policy rules. Figure 9 on page 66 shows the types of conditions and actions that are available for JUNOSe policy rules. Figure 10 on page 66 shows the types of conditions and actions that are available for PCMM policy rules.

Figure 8: JUNOS Policy Rules with Supported Conditions and Actions

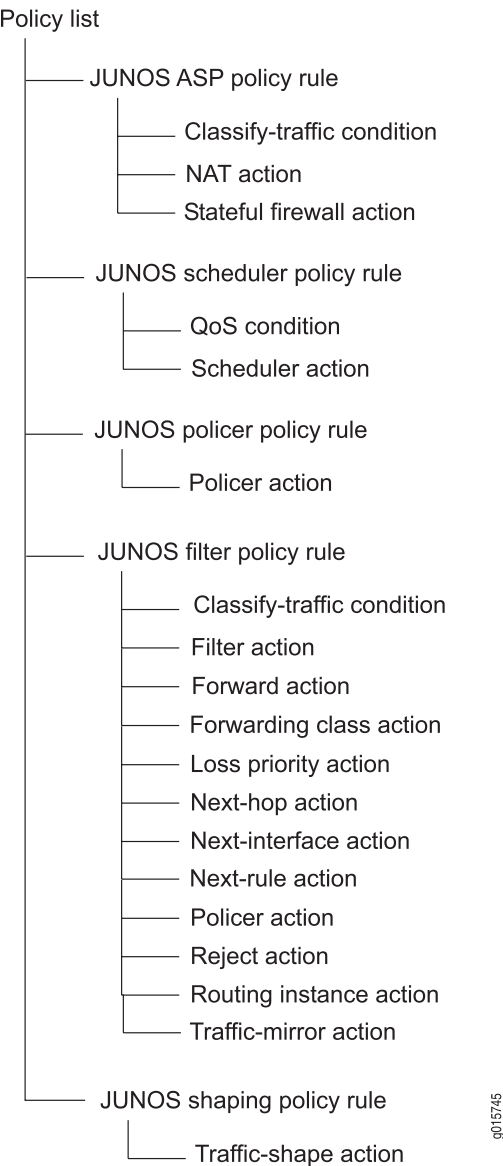
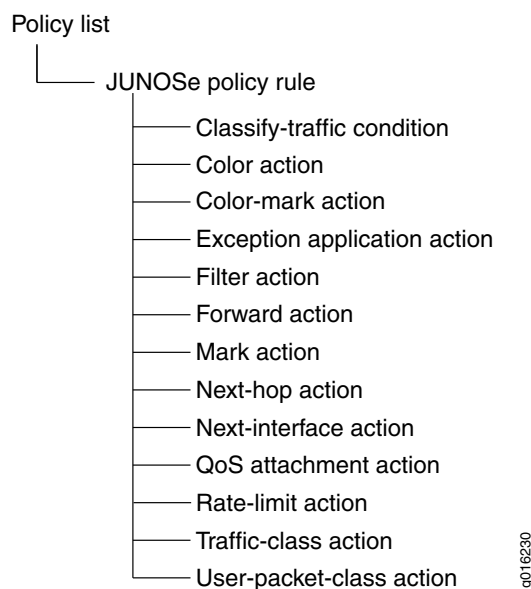
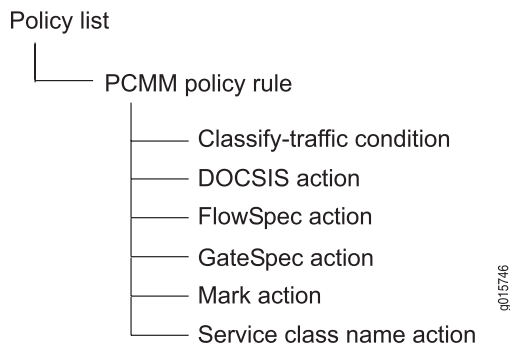


Figure 9: JUNOS Policy Rules with Supported Conditions and Actions**Figure 10: PCMM Policy Rules with Supported Conditions and Actions**

Policy Conditions

Policy conditions are values or fields that a packet must contain. If a policy rule does not contain a match condition, all packets are considered to match. There are two types of conditions:

- Classify-traffic condition—Matches can include source and destination addresses or networks; ports, packet types, IP options, TCP flags, network protocol, application protocol
- QoS condition—Matches the forwarding class of the packet

See also “PCMM Classifiers” on page 76.

Multiple Classifiers

JUNOS^e and PCMM policy rules can contain multiple classify-traffic conditions. Having multiple classifiers in a policy rule gives you more flexibility for defining services and allows you to use fewer policy rules for some applications.

If multiple policy rules have the same action, but different classify conditions, you can combine the policy rules into one policy rule. You can also set up one policy rule that has multiple classifiers, each for a different subnet or range of addresses.

If you want to collect accounting data on internal versus external traffic, you can configure one policy rule with a set of classifiers for internal traffic and one policy rule with a set of classifiers for external traffic.

Rate-Limiting with Multiple Classifiers

Multiple classifiers give you more flexibility for rate-limiting policies. Without multiple classifiers, you can rate-limit only individual traffic flows. With multiple classifiers, you can rate-limit the aggregate of traffic flows from all sources.

The following example uses multiple classifiers to rate-limit traffic to 1 Mbps for traffic going to two different subnets.

```
Policy List je-in
Policy Rule rate-limiter
ClassifyTrafficCondition CTC1
    SourceNetwork:
        any
    DestinationNetwork:
        ipAddress=172.60.40.0/0.0.0.255
ClassifyTrafficCondition CTC2
    SourceNetwork:
        any
    DestinationNetwork:
        ipAddress=172.60.20.0/0.0.0.255
Rate limit action that limits to 1 Mbps
Policy List je-out
Policy Rule forward
ClassifyTrafficCondition
    DestinationNetwork:
        any
    SourceNetwork:
        any
Forward action
```

Expanded Classifiers

For JUNOS^e policies, you can create classify-traffic conditions that the SAE expands into multiple classifiers before it installs the policy on the router. You can enter a comma-separated list of values in:

- Source network IP address, mask, and IP operation
- Destination network IP address, mask, and IP operation
- Port fields (for port-related protocols)

The software creates a classifier for each possible combination of address and port. The software does not expand classifiers for values that are entered as a range.

You would use this feature in policies that are used in IP multimedia subsystem (IMS) environments. You can also use it to simplify the configuration of JUNOS policy rules.

For example, the following classify-traffic condition has comma-separated lists for IP address, IP mask, and from port:

```
[edit policies folder f group g list l rule r traffic-condition tc]
user@host# show
source-network {
  network {
    ip-address "[192.1.1.0,192.2.1.1]";
    ip-mask "[255.255.255.0,255.255.255.255]";
    ip-operation "[1,1]";
  }
}

tcp-condition {
  protocol tcp;
  protocol-operation 1;
  source-port {
    port {
      from-port "[80,8080]";
      port-operation eq;
    }
  }
}
```

The sample classify-traffic condition is expanded into four classifiers that have the following combination of source addresses and source ports. Note that JUNOS policy rules use wildcard, not subnet, masks.

```
192.1.1.0/0.0.0.255 eq 80
192.1.1.0/0.0.0.255 eq 8080
192.2.1.1/0.0.0.0 eq 80
192.2.1.1/0.0.0.0 eq 8080
```

Policy Actions

JUNOS policy rules and PCMM policy rules can have multiple actions. JUNOS policy rules can have only one action. The types of actions available for a policy rule depend on the type of rule. See “Supported Conditions and Actions” on page 65. The following table is a description of all actions.

Table 7: Policy Actions

Action	Type of Rule	Description
Color	JUNOS	Specifies the color attribute that is applied to the packet when it passes through the router.
Color mark	JUNOS	Specifies the color-mark value that is applied to the packet when it passes through a rate-limit action.

Table 7: Policy Actions *(continued)*

Action	Type of Rule	Description
DOCSIS	PCMM	Explicitly specifies the Data over Cable Service Interface Specifications (DOCSIS) parameters of the DOCSIS service flow. It supports all DOCSIS service flow scheduling types.
Exception application	JUNOSe	Specifies an exception to the policy rule for traffic that has a specific application, such as a Web server.
Filter	JUNOS filter JUNOSe	Discards all packets that match the classify-traffic condition.
FlowSpec	PCMM	Specifies a traffic profile by using a Resource Reservation Protocol (RSVP)-style FlowSpec.
Forward	JUNOS filter JUNOSe	Forwards packets that match the classify-traffic condition; forwards packets to a particular interface and/or a next-hop address.
Forwarding class	JUNOS filter	Assigns a forwarding class to packets that match the classify-traffic condition.
GateSpec	PCMM	Specifies the session class ID in the gate. The session class ID provides a way to group gates into different classes with different authorization characteristics.
Loss priority	JUNOS filter	Assigns a packet loss priority to packets that match the classify-traffic condition.
Mark	PCMM JUNOSe	Sets the ToS field in the IP header for IPv4 packets, or sets the traffic-class field in the header for IPv6 packets to a specified value.
NAT	JUNOS ASP	Specifies the type of network address translation (source dynamic, destination static), IP address ranges, and a port range to restrict port translation when NAT is configured in dynamic-source mode.
Next hop	JUNOS filter JUNOSe	Specifies the IP address of the next hop; used to create a static route on the router; used for captive portal behavior; JUNOS filters support multiple next hops for load balancing.
Next interface	JUNOS filter JUNOSe	Defines an output interface and/or a next-hop address for a policy list; used to create a static route on the router; used for captive portal behavior.
Next rule	JUNOS filter	Causes the router to skip to and evaluate the next rule in the policy list.
Policer	JUNOS policer JUNOS filter	Specifies rate and burst size limits and the action taken if a packet exceeds those limits.
QoS attachment	JUNOSe	Specifies the QoS profile that is applied to the packet when it passes through the router.

Table 7: Policy Actions *(continued)*

Action	Type of Rule	Description
Rate limit	JUNOSe	Specifies bandwidth attributes (committed, peak, and excess rates and burst sizes) and the action taken relative to the bandwidth (filter, forward, or mark).
Reject	JUNOS filter	Discards the packet and sends an ICMP destination unreachable message to the client; can set the type of ICMP message to send.
Routing instance	JUNOS filter	Also called filter-based forwarding; directs traffic to a routing instance that is configured on the router.
Scheduler	JUNOS scheduler	Specifies transmission-scheduling and rate-control parameters. Schedulers define the priority, bandwidth, delay buffer size, rate-control status, and RED drop profiles to be applied to a particular class of traffic.
Service class name	PCMM	Specifies that traffic is controlled by a service class that is configured on the CMTS device.
Stateful firewall	JUNOS ASP	Specifies whether to filter, forward, or reject a packet. If a packet is rejected, a rejection message is returned.
Traffic class	JUNOSe	Specifies the traffic-class profile that is applied to the packet when it passes through the router.
Traffic shape	JUNOS shaping	Specifies the maximum rate of traffic transmitted on an interface.
Traffic mirror	JUNOS filter	Mirrors traffic from a destination to a source or from a source to a destination.
User packet class	JUNOSe	Specifies the user packet class that is applied to the packet when it passes through the router.

Combining Actions

JUNOS policy rules and PCMM policy rules support multiple actions. For example, in PCMM policies, you can combine a mark action with a DOCSIS parameter action, a service schedule action, or a FlowSpec action. In JUNOS policy rules you can combine the forwarding class action, routing instance action, and loss priority action. The result is that packets that match the condition are assigned to a forwarding class, directed to a routing instance on the router, and assigned a packet loss priority.

Only one of the following actions can exist in a policy rule: next-hop action, next-interface action, forward action, filter action, and reject action.

For example, if you add the next-rule action to a policy rule, do not add a next-hop action, next-interface action, forward action, filter action, or reject action to the same policy rule.

Although you can have only one action in a JUNOS policy rule, you can set up a policy list to take two corresponding actions on a packet. To do so, you create a JUNOS policy list that has more than one policy rule with the same precedence. For example, you might want a policy rule that marks a packet and a policy rule that forwards the packet to the next interface. Or you could have a policy rule that applies a traffic-class action and a policy rule that forwards the packet to the next hop.

- Related Topics**
- Policy Management Overview on page 59
 - For information about configuring and managing policies, classifiers, and actions, see Before You Configure SRC Policies on page 105
 - Example: Providing Tiered Internet Services with Policing on page 199

Dynamic Bandwidth Allocation Among Traffic Flows in SRC Policy Lists for JUNOS Routing Platforms

- Overview of Rate-Limit Hierarchies on page 71
- Parent Groups in Rate-Limit Hierarchies on page 71
- Actions in Rate-Limit Hierarchies on page 72

Overview of Rate-Limit Hierarchies

SRC support for JUNOS rate-limit hierarchies lets you configure policy lists to dynamically share unused bandwidth from high-priority traffic with lower priority traffic. A rate-limit hierarchy is a defined series of rate limits that a packet traverses within a policy list. At each level in the hierarchy, the packet is evaluated and processed as configured. Traffic preferences set in a hierarchy can specify the bandwidth allocated to a combined traffic flow and to several traffic flows within the combined flow. For example, a traffic flow may include video-on-demand traffic and Internet traffic. The video-on-demand traffic would have a high priority, but during times when not all bandwidth allocated to video-on-demand is in use, the Internet traffic can access the unused bandwidth.

Rate-limit hierarchies also enable real-time traffic to use all guaranteed bandwidth below a configured maximum. Traffic is transmitted depending on the rate limit set for the traffic flow. Preferred, high-priority traffic packets are dropped only as configured by the rate limit for that traffic flow. Lower priority traffic, however, can be dropped to keep the total traffic flow below a configured maximum limit.

Parent Groups in Rate-Limit Hierarchies

A parent group is an object that defines a rate-limit action. A parent group can contain other parent groups or be subordinate to another parent group to form a hierarchy. When you configure rate-limit hierarchies, you configure parent groups (in a policy list) that specify the rate-limit configurations and configure traffic match conditions, which reference the parent groups. You can configure a maximum of two hierarchy levels under a parent group. SRC parent groups support input, secondary-input, and output lists and can be configured for lists that have roles for JUNOS IPv4 and JUNOS IPv6.



NOTE: The SRC software does not support JUNOS external parent groups.

A parent group must contain another parent group or be referenced by a policy rule. The rate limit configured for a parent group must be configured to support hierarchical rate limits. It can also be configured to be color-aware, that is to detect ToS byte marking for red, yellow, and green. If the rate limit is color aware, a rate limit in the hierarchy can change the color of the incoming packet. An incoming packet may have the color already set by a previous rate limit or by a previous policy, such as an inbound IP or VLAN policy. Mark actions in the entire hierarchy can change the ToS values depending on the color of the packet. If the rate limit is not color aware, all packets are processed in the same way.

Actions in Rate-Limit Hierarchies

You configure the action to be taken on traffic as packets traverse a rate-limit hierarchy. At each level in a rate-limit hierarchy, traffic may:

- Drop—Drop the packet at that rate limit in the hierarchy. The packet does not change the state of any rate limit further down the hierarchy.
- Exit the hierarchy—Set the packet color and end the packet's traversal of the rate-limit hierarchy at the current rate limit. The packet is forwarded and the rate limits further down the hierarchy are not affected. Committed packets can exit the hierarchy; conformed and exceeded packets continue to the next rate limit.
- Be forwarded conditionally—Set the packet color to the result calculated by the rate limit and forward the packet to the next rate limit for processing. The next rate limit can then set the packet color and apply its actions to the packet. The **forward-conditional** option is the same as connecting the two rate limits in series.
- Be forwarded unconditionally—Set the packet color to the result calculated by the rate limit, and forward the packet to the next rate limit. After a packet is unconditionally forwarded, the packet traverses the hierarchy to the end of the hierarchy. Remaining rate limits change because packet counts are reduced. As a result, other packets may meet conformed or exceeded actions.

You can configure a rate limit to have one of the four preceding actions for committed actions and conformed actions. For exceed actions, you can also configure a rate limit to have one of first three actions in the preceding list; exceed actions do not support an option to forward packets unconditionally. The action taken depends only on the result of the rate limit, including the burst sizes, and the current bandwidth allocation. In addition, the rate limit can assign a color to the packet, depending on both the result of the rate limit and the packet's incoming color. The final color after a packet finishes traversing a

rate-limit hierarchy results from the various rate limits through which the packet passed.

- Related Topics
- Configuring Types of Actions for Rate-Limit Actions (SRC CLI) on page 178
 - For an overview of JUNOS rate-limit hierarchies, seeHierarchical Rate Limits Overview
 - Policy Information Model on page 62

Delivering QoS Services in a Cable Environment

This topic describes how SRC policies provide quality of service in the cable network environment.

Service Flow Scheduling Types

The DOCSIS protocol is used to support quality of service for traffic between the cable modem and the CMTS device. To support QoS, the DOCSIS protocol uses the concept of service flows for traffic that is transmitted between cable modems and CMTS devices. 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. Table 8 on page 73 describes the service flow scheduling types and the QoS parameters that you can set for each type.

The SRC software is compliant with the service flow scheduling types as defined in the PacketCable Multimedia Specification PKT-SP-MM-I03-051221. See the specification for detailed information about each scheduling type.

Table 8: DOCSIS Service Flow Scheduling Types

Type	Description	Suitable Traffic Type(s)	QoS Parameters
Best effort	For upstream service flows. The CMTS scheduler grants transmit opportunities on a first-come first-served basis. You can supplement best effort with QoS parameters.	Standard Internet traffic such as Web browsing, e-mail, or instant messaging	Traffic priority Request transmission policy Maximum sustained traffic rate Maximum traffic burst Minimum reserved traffic rate Assumed minimum reserved-traffic-rate packet size

Table 8: DOCSIS Service Flow Scheduling Types *(continued)*

Type	Description	Suitable Traffic Type(s)	QoS Parameters
Non-real-time polling service (NRTPS)	<p>For upstream service flows.</p> <p>The CMTS scheduler sends unicast polls to cable modems on a fixed interval to determine whether data is queued for transmission on a particular service flow. If data is queued, the scheduler provides a transmission grant for the service flow.</p>	Standard Internet traffic that requires high throughput, and traffic that requires variable-sized data grants on a regular basis, such as high-bandwidth FTP.	<p>Traffic priority</p> <p>Request transmission policy</p> <p>Maximum sustained traffic rate</p> <p>Maximum traffic burst</p> <p>Minimum reserved traffic rate</p> <p>Assumed minimum reserved-traffic-rate packet size</p> <p>Nominal polling interval</p>
Real-time polling service (RTPS)	<p>For upstream service flows.</p> <p>Analogous to NRTPS, except that the fixed polling interval is typically very short.</p> <p>Offers request opportunities that meet the service flows' real-time needs and allows the cable modem to specify the size of the desired grant.</p>	<p>Real-time traffic that generates variable-sized data packets on a periodic basis and has inflexible latency and throughput requirements.</p> <p>Applications include Moving Pictures Experts Group (MPEG) video.</p>	<p>Request transmission policy</p> <p>Maximum sustained traffic rate</p> <p>Maximum traffic burst</p> <p>Minimum reserved traffic rate</p> <p>Assumed minimum reserved-traffic-rate packet size</p> <p>Nominal polling interval</p> <p>Tolerated poll jitter</p>
Unsolicited grant service (UGS)	<p>For upstream service flows.</p> <p>The CMTS device provides a fixed-size grant to a service flow at fixed intervals without additional polling or interaction. UGS eliminates much of the overhead associated with the polling flow types.</p>	<p>Real-time traffic that generates fixed-size data packets on a periodic basis.</p> <p>Applications include voice over IP (VoIP)</p>	<p>Request transmission policy</p> <p>Unsolicited grant size</p> <p>Grants per interval</p> <p>Nominal grant interval</p> <p>Tolerated grant jitter</p>
Unsolicited grant service with activity detection (UGS-AD)	<p>For upstream service flows.</p> <p>A hybrid of the UGS and RTPS scheduling types.</p> <ul style="list-style-type: none"> ■ When there is activity, the CMTS device sends unsolicited fixed grants at fixed intervals to the cable modem. ■ When there is no activity, the CMTS device sends unicast poll requests to the cable modem to conserve unused bandwidth. 	Applications include voice activity detection, also known as silence suppression	<p>Request transmission policy</p> <p>Nominal polling interval</p> <p>Tolerated poll jitter</p> <p>Unsolicited grant size</p> <p>Grants per interval</p> <p>Nominal grant interval</p> <p>Tolerated grant jitter</p>

Table 8: DOCSIS Service Flow Scheduling Types *(continued)*

Type	Description	Suitable Traffic Type(s)	QoS Parameters
Downstream	For downstream service flows. 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.	All downstream traffic	Traffic priority Maximum sustained traffic rate Maximum traffic burst Minimum reserved traffic rate Assumed minimum reserved-traffic-rate packet size Maximum latency

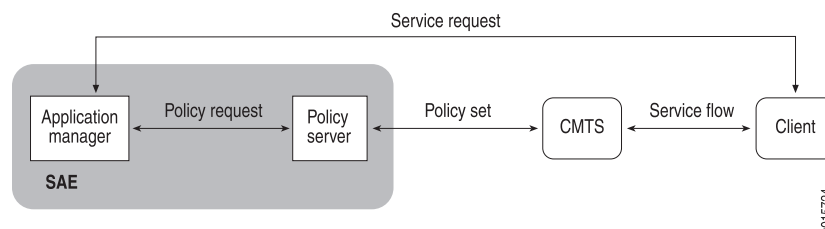
Client Type 1 Support

The PCMM specification defines three types of clients, and defines a client as a logical entity that can send or receive data. The SRC software supports client type 1, which represents endpoints such as PC applications or gaming consoles that lack specific QoS awareness or signaling capabilities. Client type 1 entities communicate with an application manager to request service, and the CMTS device manages the QoS signaling.

Client type 1 entities support the proxied QoS with policy push scenario of service delivery defined in the 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.

Proxied QoS with Policy Push

In the proxied QoS with policy push scenario of service delivery, the client requests a service by sending a service request to the application manager. The application manager determines the QoS needs of the request and sends a policy request to the policy server. The policy server validates the policy request and if the decision is affirmative, sends a policy set message to the CMTS device. The CMTS device performs admission control on the requested QoS envelope, installs the policy decision, and establishes the service flow to the client with the requested QoS levels.

Figure 11: Authorization Framework for Proxied QoS with Policy Push

PCMM Gate

A PCMM gate is a logical representation of a policy decision that has been installed on the CMTS device. The gate performs traffic classification and enforces QoS policies on media streams.

The set of service flow characteristics that provide enhanced QoS is the envelope. A CMTS gate contains up to three envelopes that indicate authorized, reserved, and committed resources for the service flow that corresponds to the gate. A gate defines a resource authorization envelope that consists of IP-level QoS parameters as well as classifiers that define the scope of service flows that can be established against the gate.

Three elements of a gate discussed here are session class ID, classifiers, and traffic profiles.

Session Class ID

The session class ID provides a way for the application manager and the policy server to group gates into classes with different authorization characteristics. A CMTS device can perform authorization based not only on the requested QoS and the gate's authorized flow specification (FlowSpec), but also on the session class ID specified in the GateSpec. For example, you could use the session class ID to represent a prioritization scheme that allows either the policy server or the CMTS device to preempt a preauthorized gate in favor of allowing a new gate with a higher priority to be authorized.

Use the GateSpec action to specify the session class ID for a gate.

PCMM Classifiers

The classifier identifies the IP flow that will be mapped to the DOCSIS service flow associated with the gate. You define the classifier by using a classify-traffic condition.

PCMM Classifiers and Extended Classifiers

Classify-traffic conditions comply with the classifiers specified in PacketCable Multimedia Specification PKT-SP-MM-I02-040930 (referred to as PCMM I02) as well as the extended classifiers in PacketCable Multimedia Specification PKT-SP-MM-I03-051221 (referred to as PCMM I03).

To specify which version of the PCMM classifiers that you are using, see one of the following:

- Specifying the PCMM Classifier Type (SRC CLI) on page 116
- Configuring Policy Lists (C-Web Interface)
- Adding a Policy Rule
- Specifying the PCMM Classifier Type (C-Web Interface)

PCMM I02 classifiers do not support IP masks or a range of port numbers. PCMM I03 classifiers do support IP masks and a range of port numbers.

You define classifiers for PCMM irrespective of whether the policy is meant for I02 or I03. At service activation time, depending on whether the SAE is configured to use I02 or I03 policies, the policy engine does the appropriate translations. For example, if I02 policies are to be used, source and destination IP masks and ranges of port numbers are ignored.

You can configure all fields for extended PCMM classifiers (PCMM I03), except for classifierID, activation state, and action. At service activation, the policy engine sets these fields as follows:

- ClassifierID = A system-generated number
- Activation state = Active
- Action = Add

Guidelines for Configuring Classifiers

When you configure classify-traffic conditions for PCMM policies, keep in mind the following:

- Do not leave the IP address field empty.
- For PCMM classify-traffic conditions, there are two special protocol values:
 - 256 matches traffic that has any IP protocol value
 - 257 matches both TCP and UDP traffic
- PCMM I02 classifiers do not support IP masks or a range of port numbers.
- PCMM I03 classifiers to support IP masks and a range of port numbers.

Traffic Profiles

There are three ways to express the traffic profile for a gate:

- DOCSIS parameters—Specifies the traffic profile through DOCSIS-specific parameters.
- Service class name—Name of a service class that is configured on the CMTS device.
- FlowSpec—Defines the traffic profile through an RSVP-like parameterization scheme.

You can also mark the ToS byte of a packet as it gets to the gate.

DOCSIS Parameters

You use DOCSIS parameters in a network that uses version 1.1 of the DOCSIS protocol. To define DOCSIS parameters for a traffic profile, use the DOCSIS action. This action supports service flow scheduling types and QoS parameters described in

“Delivering QoS Services in a Cable Environment” on page 73. See one of the following:

- “Configuring DOCSIS Actions (SRC CLI)” on page 151 .
- Configuring DOCSIS Actions (C-Web Interface)

Service Class Name

To use a service class name for a traffic profile, use the service class name action. Instead of setting QoS parameters, you specify the name of a service class that is configured on the CMTS device. See one of the following:

- Configuring Service Class Name Actions (SRC CLI) on page 185
- Configuring Service Class Name Actions (C-Web Interface)

FlowSpec Parameters

You can use an RSVP-style FlowSpec to specify a traffic profile. A FlowSpec is made up of two parts, a traffic specification (TSpec) and a service request specification (RSpec). The TSpec describes the traffic requirements for the flow, and the RSpec specifies resource requirements for the desired service.

TSpec parameters defined in the FlowSpec are:

- Bucket rate
- Bucket depth
- Peak rate
- Minimum policed unit
- Maximum packet size

RSpec parameters defined in the FlowSpec are:

- Reserved rate
- Slack term

Types of FlowSpec Services

FlowSpecs support two types of services—controlled load and guaranteed.

- Controlled-load service can be used to provide minimum bandwidth guarantees, and is suitable for applications that are not latency sensitive. Controlled-load service allows applications to have low delay and high throughput even during times of congestion. Controlled-load service can be closely approximated to the best-effort service flow scheduling type. Controlled-load services support TSpec parameters only.
- Guaranteed service allows applications to reserve bandwidth, and is suitable for latency and jitter-sensitive applications such as voice, MPEG video, or gaming. The CMTS device uses the traffic profile parameters specified in the FlowSpec

to select one of the two types of DOCSIS scheduling types that can provide guaranteed services—RTPS and UGS. Guaranteed services support both TSpec and RSpec parameters.

Table 9 on page 79 shows how the FlowSpec service types map to the DOCSIS service scheduling types.

Table 9: Mapping FlowSpec Types

FlowSpec Service Type	DOCSIS Scheduling Type	Application Example
Guaranteed	Unsolicited Grant Service (UGS)	Voice over IP
Guaranteed	Real-Time Polling Service (RTPS)	Guaranteed VPN
Controlled load	Best effort	Standard Internet service

FlowSpec Parameters

Table 10 on page 79 shows the parameters that you can set for each service type.

Table 10: Parameters Available for Each Type of Service

Controlled Load	Guaranteed Service
Token bucket rate	Token bucket rate
Token bucket size	Token bucket size
Peak data rate	Peak data rate
Minimum policed unit	Minimum policed unit
Maximum packet size	Maximum packet size
	Rate
	Slack term

Marking Packets

You can also mark packets and then install policies on the router that handle the marked packets in a certain way. The mark action causes the ToS byte to be set in the IP header of IPv4 traffic or the traffic-class field to be set in the IP header of IPv6 traffic. For example, to offer videoconferencing, you could:

1. Create a classify-traffic condition that causes the CMTS device to classify the traffic.

2. Create a mark action that causes the CMTS device to mark the ToS byte or traffic-class field in the classified traffic.
3. Create a policy on the router that classifies the traffic according to the marked ToS byte.

- Related Topics**
- Policy Management Overview on page 59
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring QoS Conditions (SRC CLI) on page 147
 - Example: Providing Premium Services on page 203

Chapter 5

Overview of Using Local and Global Parameters

- Overview of Global and Local Parameters on page 81
- Parameter Types on page 82

Overview of Global and Local Parameters

Policy definitions are templates that the policy engine uses to construct policies that the SAE installs on the router or provisions on the CMTS device. When you configure the policy template, you can assign parameter values. Before it creates a policy for installation on the router, the policy engine substitutes parameter values with specific values. The policy engine uses the parameter value acquisition process to obtain the specific values.

Policies can use global or local parameters:

- Global parameters—Are available to use in any policy. With global parameters, you can define parameters once and then reuse them in many policies. Typically, global parameters are not changed often, and if changes are necessary, local parameters are used.
- Local parameters—Are available only for the policy group in which the parameter is defined.

The SRC software provides many predefined built-in parameters and runtime parameters. Runtime parameters are built-in parameters that are filled in with an actual value from the running system when the policy is installed on the router. For example, the `interface_speed` parameter is filled in with the actual speed of the router interface. You cannot change the values of built-in or runtime parameters.

Related Topics

- Parameters and Substitutions on page 209.
- Configuring Global Parameters (SRC CLI) on page 101
- Configuring Local Parameters (SRC CLI) on page 103
- Viewing Predefined Global Parameters (SRC CLI) on page 101
- Parameter Types on page 82

Parameter Types

Global and local parameters are assigned a type. The type indicates the SRC CLI options in which you can use the parameter.

For example, address is a type of parameter. In the SRC CLI, whenever there is an option for which you can specify an IP address, you can use the ? to display a list of all local and global parameters of type address. For example:

```
user@host# set source-network network ip-address ?
```

Possible completions:

<ip-address> IP address of the source or destination network or host

gateway_ipAddress

interface_ipAddress

service_ipAddress

user_ipAddress

virtual_ipAddress

There are a few cases in which a global parameter value appears, but because of the context, the value does not make sense to use. For example, in NAT actions, the global parameter any appears in for the IP network setting. In this context, any is not a valid value.

Table 11 on page 82 lists the parameter types, the predefined parameters for each type, the policy objects in which you can use the parameter type, and how the type is used.

Table 11: Parameter Types

Type	Predefined Parameters	Used In	Used to Specify
address	gateway_ipAddress	Classify-traffic condition	IP addresses in dotted decimal notation.
	interface_ipAddress		
	service_ipAddress	Next-interface action	
	user_ipAddress	Next-hop action	
	virtual_ipAddress		

Table 11: Parameter Types (continued)

Type	Predefined Parameters	Used In	Used to Specify
addressMask	interface_ipMask	Classify-traffic condition	IP masks in dotted decimal notation.
	service_ipMask		
	user_ipMask		For JUNOS policies and JUNOS policies (except for firewall policies), a mask must be equivalent to some prefix length. For example, 255.255.255.0 is allowed, but 255.255.255.1 is not. The software searches this constraint for default parameter values, but not for any other substitution values until runtime when the policy engine constructs the policy.
allowIpOptions		Classify-traffic condition	
any			The set of all values.
applicationProtocol	bootp, dce_rpc, dce_rpc_portmap, dns, exec, ftp, h323, green, icmp_app, iiop, netbios, netshow, realaudio, rpc, rpc_portmap, rtsp, shell, snmp, sqlnet, tftp, traceroute, winframe, yellow	Classify-traffic condition (Predefined parameters map protocol numbers to synonyms.)	
bandwidthSizeUnit	bps	Policer action	
	percent		
boolean	false		
	true		
burst		Rate-limit action	Burst sizes. The range is $2^{14} - 2^{32} - 1$.
		Policer action	
		DOCSIS action	
color	green	Classify-traffic condition	Color of action or classifier.
	red	Color action	The policy engine validates these values; the substitution engine does not.
	yellow		
dceRpcUuid		Classify-traffic condition	

Table 11: Parameter Types *(continued)*

Type	Predefined Parameters	Used In	Used to Specify
dropProfileProtocol	any_protocol	Scheduler action	
	non_tcp		
	tcp_only		
dropProfileType	interpolated	Scheduler action	
	segmented		
exceptionApplication	http	Exception application actions	The policy rule for traffic that has a specific application, such as a Web server
forwardingClass		Classify-traffic condition	
		QoS condition	
fragOffset		Classify-traffic condition	<p>The value of the fragment offset field of IP packets.</p> <p>For JUNOS routers:</p> <ul style="list-style-type: none"> ■ eq 0—Equal to 0 ■ eq 1—Equal to 1 ■ gt 1—Greater than 1 ■ any—Any value <p>For JUNOS routing platforms and PCMM policies, integer in the range 0–8191.</p> <p>The policy engine validates these values; the substitution engine does not.</p>
grantSize		DOCSIS action	
icmpCode		Classify-traffic condition	8-bit values that represent patterns in the ICMP code and ICMP type fields in IP packets. The policy engine validates these values; the substitution engine does not.
icmpType			
igmpType		Classify-traffic condition	8-bit values that represent patterns in the IGMP type field in IP packets. The policy engine validates these values; the substitution engine does not.

Table 11: Parameter Types (continued)

Type	Predefined Parameters	Used In	Used to Specify
interfaceGroup		Classify-traffic condition	
InterfaceSpec	bfwlf gfwlf	Next-interface action	The router interface. For JUNOS interfaces, the format is: ‘< type of specifier > = < value >’ For example: name = ‘fastEthernet3/0’ For JUNOS interfaces, the format is: ‘name = < mediatype > - < slot > / < pic > / < port > . < unit >’ For example: ‘name = AT-0/1/0.0’
interval		DOCSIS action	
ipFlags		Classify-traffic condition	3-bit values that represent patterns for the IP flags field in an IP packet. The high bit is reserved, the middle bit is don’t fragment, and the low bit is more fragments.
ipFlagsMask			
ipSecSpi		Classify-traffic condition	
IPv4range			
jitter		DOCSIS action	
l2cAccessMode	l2c_access_mode		
l2cAccessStreamRate	l2c_access_downstream_rate, l2c_access_upstream_rate, l2c_attainable_downstream_rate, l2c_attainable_upstream_rate, l2c_maximum_downstream_rate, l2c_maximum_upstream_rate, l2c_min_downstream_rate, l2c_min_upstream_rate, l2c_minimum_low_power_downstream_rate, l2c_minimum_low_power_upstream_rate		
l2cDelay	l2c_actual_interleaving_downstream_delay, l2c_actual_interleaving_upstream_delay, l2c_maximum_interleaving_downstream_delay, l2c_maximum_interleaving_upstream_delay		
l2cDslStatus	l2c_dsl_line_stat		

Table 11: Parameter Types *(continued)*

Type	Predefined Parameters	Used In	Used to Specify
matchDirection	both input output	Classify-traffic condition	
maxLatency		DOCSIS action	
messageType		Reject action	
microSecond			
natTranslationType		NAT action	
network	any	Classify-traffic condition NAT action	<p>IP subnets using two forms:</p> <p>< address > / < mask ></p> <p>< address > / < prefixLength ></p> <p>where < address > and < mask > are in the traditional dotted decimal notation.</p> <p>< prefixLength > is a number in the range 0–32, which specifies how many of the first bits in the address specify the network.</p> <p>In policy conditions, network specifies patterns for the address fields in packets. Networks can be preceded by “not” to indicate that the condition matches every address not in the subnet.</p>
networkOperation		Classify-traffic condition	<p>Whether a network field of a packet should match or not match the value specified in a policy condition.</p> <ul style="list-style-type: none"> ■ 0—Does not match ■ 1—Matches
packetLength		Classify-traffic condition DOCSIS action FlowSpec action	

Table 11: Parameter Types *(continued)*

Type	Predefined Parameters	Used In	Used to Specify
packetLossPriority	any_priority high_priority low_priority	Loss priority action	
packetOperation		Rate-limit action Policer action Stateful firewall	<p>Actions taken on packets.</p> <p>For rate-limit actions, valid values are: '\$forward', '\$filter', and '\$mark < tosByte > < tosMask >'.</p> <p>For policer actions, valid values are: filter, forwardingClass, lossPriority.</p> <p>For stateful firewalls, valid values are: filter, forward, reject.</p> <p>The policy engine validates these values; the substitution engine does not.</p>
percent		Scheduler action	
policedUnit		FlowSpec action	
port	service_port	Classify-traffic condition NAT action	16-bit values that represent patterns in the port fields in IP packets.
portOperation	eq neq	Classify-traffic condition	<p>Whether a port field should match or not match the value(s) specified in a condition. For JUNOS policies valid values are: '\$eq', '\$lt', '\$gt', '\$neq' and '\$range'.</p> <p>For JUNOS the allowed values are:</p> <ul style="list-style-type: none"> ■ 0—Does not match ■ 1—Matches <p>The policy engine validates these values; the substitution engine does not.</p>
prPrecedence		Policy rule	

Table 11: Parameter Types *(continued)*

Type	Predefined Parameters	Used In	Used to Specify
protocol	ah, egp, esp, gre, icmp, igmp, ip, ipip, ospf, pim, rsvp, tcp, udp	Classify-traffic condition (Predefined parameters map protocol numbers to synonyms.)	8-bit values that represent patterns in the protocol field in IP packets. The policy engine validates these values; the substitution engine does not.
protocolOperation	is not	Classify-traffic condition	Whether a protocol field of a packet should match or not match the value specified in a policy condition. <ul style="list-style-type: none"> ■ 0—Does not match ■ 1—Matches
qosProfileSpec		QoS-attachment action	Strings in QoS attachment actions that specify QoS profiles. They can be any string that names a QoS profile on the JUNOS router.
rate	interface_speed	Rate-limit action Policer action DOCSIS action FlowSpec action Traffic-shape action	Rates in the range $0-2^{32}-1$.
rateLimitType	one_rate two_rate	Rate-limit action	Rate-limit type. The allowed values are '\$one-rate' and '\$two-rate'. The policy engine validates these values; the substitution engine does not.
requestTransmissionPolicy		DOCSIS action	
routingInstance		Routing instance action	
rpcProgramNumber		Classify-traffic condition	
schedulerBufferSize		Scheduler action	
schedulerBufferSizeUnit	buffer_size_percentage buffer_size_remainder temporal	Scheduler action	

Table 11: Parameter Types *(continued)*

Type	Predefined Parameters	Used In	Used to Specify
schedulerPriority	high	Scheduler action	
	low		
	medium_high		
	medium_low		
	strict_high		
schedulerTransmitRate		Scheduler action	
schedulerTransmitRateUnit	rate_in_bps	Scheduler action	
	rate_in_percentage		
	rate_in_remainder		
serviceClassName		Service class name action	
serviceNumber	controlled_load_service	FlowSpec action	
	guaranteed_service		
sessionClassIdPriority		GateSpec action	
slackTerm		FlowSpec action	
snmpCommand	get	Classify-traffic condition	
	get_next		
	set		
	trap		
tcpFlags		Classify-traffic condition	6-bit values that represent patterns for the TCP flags field in IP packets. The bits from high to low mean: urgent, acknowledge, push, reset, synchronize, finish.
tcpFlagsMask			
timeout		Classify-traffic condition	
tokenBucketSize		FlowSpec action	

Table 11: Parameter Types *(continued)*

Type	Predefined Parameters	Used In	Used to Specify
tosByte		Classify-traffic condition	8-bit values that represent patterns in the ToS byte field in IP packets.
tosByteMask		Rate-limit action Mark action	When tosByteMask is used in ToS conditions, the allowed values are 0, 224, 252, and 255. The policy engine validates these values; the substitution engine does not.
traceRouteTtlThreshold		Classify-traffic condition	
trafficClassSpec		Traffic-class action	Strings in traffic-class actions that specify traffic-class profiles. They can be any string that names a traffic class on the JUNOS router.
trafficPriority		DOCSIS action	
trafficProfileType	best_effort unsolicited_grant down_stream unsolicited_grant_with_activity_detection real_time non_real_time	DOCSIS action	Service flow scheduling type
translationType			
userPacketClass		User packet class action	4-bit value. For JUNOS policies, valid values are in the range 0–15. The policy engine validates these values; the substitution engine does not.

Predefined Global Parameters

Table 12 on page 91 describes the predefined built-in and runtime global parameters that the SRC software provides. Only three of the predefined parameters can be modified: any, bwlf, and gfwlf.

Table 12: Predefined Global Parameters

Predefined Parameter	Description	Type	Runtime
ah	Maps protocol 51 to AH	protocol	
any	This network matches any address	network	
any_priority	Sets packet loss priority to “any”	packetLossPriority	
any_protocol	Sets drop profile protocol to “any”	dropProfileProtocol	
best_effort	Sets the service flow scheduling type to best effort	trafficProfileType	
bfwlf	Specifier of the interface that leads to the bronze firewall server	interfaceSpec	Yes
bootp	Specifies the BOOTP protocol	applicationProtocol	
both	Specifies the direction of the policy as input and output	matchdirection	
bps	Specifies that the indicated bandwidth size is in bps	bandwidthSizeUnit	
buffer_size_percentage	Specifies that the indicated buffer size is a percentage	schedulerBufferSizeUnit	
buffer_size_remainder	Specifies that the indicated buffer size is a remainder	schedulerBufferSizeUnit	
controlled_load_service	Specifies that the type of FlowSpec service is controlled-load service	serviceNumber	

Table 12: Predefined Global Parameters *(continued)*

Predefined Parameter	Description	Type	Runtime
dce_rpc	Specifies the DCE RPC protocol	applicationProtocol	
dce_rpc_portmap	Specifies the DCE RPC portmap	applicationProtocol	
dns	Specifies the DNS protocol	applicationProtocol	
down_stream	Sets the service flow scheduling type to downstream	trafficProfileType	
egp	Maps protocol 8 to EGP	protocol	
eq	Matches packets with a port that is equal to the specified port	portOperation	
esp	Maps protocol 50 to ESP	protocol	
exec	Specifies the Exec protocol	applicationProtocol	
false	Sets Boolean values to false	boolean	
ftp	Specifies the FTP protocol	applicationProtocol	
gateway_ipAddress	IP address of the gateway as specified by the service object	address	Yes
get	Specifies the get SNMP command	snmpCommand	
get_next	Specifies the get-next SNMP command	snmpCommand	
gfwlf	Specifier of the interface that leads to gold firewall server	interfaceSpec	Yes

Table 12: Predefined Global Parameters (continued)

Predefined Parameter	Description	Type	Runtime
gre	Maps protocol 47 to GRE	protocol	
green	Specifies the color that indicates a low drop preference	color	Yes
guaranteed	Specifies that the type of FlowSpec service is guaranteed service	serviceNumber	
h323	Specifies the H.323 protocol	applicationProtocol	
high	Sets the scheduler priority to high	schedulerPriority	
high_priority	Sets the packet loss priority (PLP) to high	packetLossPriority	
icmp	Maps protocol 1 to ICMP	protocol	
icmp_app	Specifies the ICMP protocol	applicationProtocol	
igmp	Maps protocol 2 to IGMP	protocol	
iiop	Specifies the Internet Inter-ORB Protocol, a TCP protocol	applicationProtocol	
input	Specifies the direction of the policy as input	matchdirection	
interface_ipAddress	IP address of the interface	address	Yes
interface_ipMask	IP mask of the interface	addressMask	Yes

Table 12: Predefined Global Parameters *(continued)*

Predefined Parameter	Description	Type	Runtime
interface_speed	Speed of the subscriber's IP interface on the router or the speed of the subscriber's DOCSIS interface	rate	
interpolated	Sets the drop profile type to interpolate	dropProfileType	
ip	Maps protocol 0 to IP	protocol	
ipip	Maps protocol 4 to IP-IP	protocol	
is	Matches packets with the protocol that is equal to the specified protocol	protocolOperation	
l2c_access_downstream_rate	L2C downstream rate	l2cAccessStreamRate	Yes
l2c_access_mode	L2C access mode	l2cAccessMode	Yes
l2c_access_upstream_rate	L2C upstream rate	l2cAccessStreamRate	Yes
l2c_actual_interleaving_downstream_delay	L2C interleaving downstream delay	l2cDelay	Yes
l2c_actual_interleaving_upstream_delay	L2C interleaving upstream delay	l2cDelay	Yes
l2c_attainable_downstream_rate	L2C attainable downstream rate	l2cAccessStreamRate	Yes
l2c_attainable_upstream_rate	L2C attainable upstream rate	l2cAccessStreamRate	Yes
l2c_dsl_line_stat	L2C DSL line status	l2cDslStatus	Yes

Table 12: Predefined Global Parameters (continued)

Predefined Parameter	Description	Type	Runtime
l2c_maximum_downstream_rate	L2C maximum downstream rate	l2cAccessStreamRate	Yes
l2c_maximum_interleaving_downstream_delay	L2C maximum interleaving downstream delay	l2cDelay	Yes
l2c_maximum_interleaving_upstream_delay	L2C maximum interleaving upstream delay	l2cDelay	Yes
l2c_maximum_upstream_rate	L2C maximum upstream rate	l2cAccessStreamRate	Yes
l2c_min_downstream_rate	L2C minimum downstream rate	l2cAccessStreamRate	Yes
l2c_min_upstream_rate	L2C minimum upstream rate	l2cAccessStreamRate	Yes
l2c_minimum_low_power_downstream_rate	L2C minimum low power downstream rate	l2cAccessStreamRate	Yes
l2c_minimum_low_power_upstream_rate	L2C minimum low power upstream rate	l2cAccessStreamRate	Yes
low	Sets scheduler priority to low	schedulerPriority	
low_priority	Sets packet loss priority to low	packetLossPriority	
medium_high	Sets scheduler priority to medium-high	schedulerPriority	
medium_low	Sets scheduler priority to medium-low	schedulerPriority	
neq	Matches packets with a port that is not equal to the specified port	portOperation	

Table 12: Predefined Global Parameters *(continued)*

Predefined Parameter	Description	Type	Runtime
netbios	Specifies the NetBIOS protocol	applicationProtocol	
netshow	Specifies the NetShow protocol	applicationProtocol	
non_real_time	Sets the service flow scheduling type to NRTPS	trafficProfileType	
non_tcp	Sets the drop profile protocol to any protocol other than TCP	dropProfileProtocol	
not	Matches packets with the protocol that is not equal to the specified protocol	protocolOperation	
one_rate	Sets the rate-limit type to one rate	rateLimitType	
ospf	Maps protocol 89 to OSPF	protocol	
output	Specifies the direction of the policy as output	matchdirection	
percent	Specifies that the indicated bandwidth size is a percentage of bandwidth	bandwidthSizeUnit	
pim	Maps protocol 103 to PIM	protocol	
rate_in_bps	Specifies that the indicated transmit rate is in bps	schedulerTransmitRateUnit	
rate_in_percentage	Specifies that the indicated transmit rate is a percentage	schedulerTransmitRateUnit	

Table 12: Predefined Global Parameters (continued)

Predefined Parameter	Description	Type	Runtime
rate_in_remainder	Specifies that the indicated transmit rate is a remainder	schedulerTransmitRateUnit	
realaudio	Specifies the RealAudio protocol	applicationProtocol	
real_time	Sets the service flow scheduling type to RTPS	trafficProfileType	
red	Specifies the color that indicates a high drop preference	color	Yes
rpc	Specifies the RPC UDP or TCP protocols	applicationProtocol	
rpc_portmap	Specifies the RPC portmap protocol	applicationProtocol	
rsvp	Maps protocol 46 to RSVP	protocol	
rtsp	Specifies the Real-Time Streaming Protocol	applicationProtocol	
sctp	Maps protocol 132 to the Stream Control Transmission Protocol	protocol	
segmented	Sets the drop profile type to segmented	dropProfileType	
service_ipAddress	IP address of the service as specified by the service object	address	Yes
service_ipMask	IP mask of the service as specified by the service object	address	Yes

Table 12: Predefined Global Parameters *(continued)*

Predefined Parameter	Description	Type	Runtime
service_port	Service port as specified by the service object	port	Yes
set	Specifies the set SNMP command	snmpCommand	
shell	Specifies the Shell protocol	applicationProtocol	
snmp	Specifies the SNMP protocol	applicationProtocol	
sqlnet	Specifies the SQLNet protocol	applicationProtocol	
strict_high	Sets scheduler priority to strict-high	schedulerPriority	
tcp	Maps protocol 6 to TCP	protocol	
tcp_only	Sets the drop profile protocol to TCP	dropProfileProtocol	
temporal	Specifies that the indicated buffer size is temporal	schedulerBufferSizeUnit	
tftp	Specifies the Trivial File Transfer Protocol	applicationProtocol	
traceroute	Specifies the Traceroute protocol	applicationProtocol	
trap	Specifies the trap SNMP command	snmpCommand	
true	Sets the Boolean value to true	boolean	
two_rate	Sets the rate-limit type to two rate	rateLimitType	

Table 12: Predefined Global Parameters *(continued)*

Predefined Parameter	Description	Type	Runtime
udp	Maps protocol 17 to UDP	protocol	
unsolicited_grant	Sets the service flow scheduling type to UGS	trafficProfileType	
unsolicited_grant_with_activity_detection	Sets the service flow scheduling type to UGS-AD	trafficProfileType	
user_ipAddress	IP address of the subscriber	address	Yes
user_ipMask	IP mask of the subscriber	address	Yes
virtual_ipAddress	Virtual portal address of the SSP that is used in redundant SAE installations	address	Yes
winframe	Specifies the WinFrame protocol	applicationProtocol	
yellow	Specifies the color that indicates a medium drop preference	color	Yes

Naming Global Parameters

A global parameter is stored in the directory with the parameter name as its naming attribute. The directory stores the case for the parameter name; however, the directory does not allow you to create another global parameter with a name that differs only by the use of upper and lowercase letters. For example, if there is a parameter named fastspeed, the directory will not allow the creation of a parameter named fastSpeed without first deleting fastspeed.

Also, when you define a substitution for a global parameter, make sure that the case in the substitution matches the case of the global parameter.

- Related Topics**
- Overview of Global and Local Parameters on page 81
 - Configuring Global Parameters (SRC CLI) on page 101
 - Configuring Local Parameters (SRC CLI) on page 103
 - Viewing Predefined Global Parameters (SRC CLI) on page 101
 - Viewing Runtime Parameters (SRC CLI) on page 104

Chapter 6

Configuring Local and Global Parameters (SRC CLI)

- Viewing Predefined Global Parameters (SRC CLI) on page 101
- Configuring Global Parameters (SRC CLI) on page 101
- Configuring Local Parameters (SRC CLI) on page 103
- Viewing Runtime Parameters (SRC CLI) on page 104

Viewing Predefined Global Parameters (SRC CLI)

Purpose View predefined global parameters.

Action user@host> **show configuration policies global-parameters**
?
Possible completions:
 <name> Parameter name
 any Parameter name
 bfwIf Parameter name
 fc_assured Parameter name
 fc_besteffort Parameter name
 fc_expedited Parameter name
 fwEnterpriseMaxPriority
 Parameter name
 fwEnterpriseMinPriority
 Parameter name
 fwMaxPriority Parameter name
 fwMinPriority Parameter name
 gfwIf Parameter name

- Related Topics**
- Overview of Global and Local Parameters on page 81
 - Configuring Global Parameters (SRC CLI) on page 101
 - Viewing Predefined Global Parameters (C-Web Interface)
 - Parameter Types on page 82

Configuring Global Parameters (SRC CLI)

If you change global variables for policies, the change takes effect the next time a service is activated; the change does not take effect for active service sessions.

Use the following configuration statement to create a global parameter:

```
policies global-parameters name {
  description description ;
  default-value default-value ;
  type type ;
}
```

To create a global parameter:

1. From configuration mode, enter the global parameter configuration. For example, to create a parameter called bandwidth:

```
user@host# edit policies global-parameters bandwidth
```

2. (Optional) Enter a description for the parameter. You can provide extra information and examples of how the parameter is used.

```
[edit policies global-parameters bandwidth]
user@host# set description description
```

3. (Optional) Configure a default value that the policy engine uses if no other values are provided during the parameter value acquisition process.

See “Parameter Types” on page 82 for valid values of each parameter type.

```
[edit policies global-parameters bandwidth]
user@host# set default-value default-value
```

4. (Optional) Type of attribute for which you can use the parameter.

```
[edit policies global-parameters bandwidth]
user@host# set type type
```

5. (Optional) Verify your configuration.

```
[edit policies global-parameters bandwidth]
user@host# show
default-value 5000000;
type rate;
```

- Related Topics**
- Overview of Global and Local Parameters on page 81
 - Configuring Global Parameters (C-Web Interface)
 - Configuring Local Parameters (SRC CLI) on page 103
 - Viewing Predefined Global Parameters (SRC CLI) on page 101
 - Defining Parameters for the SRC Software on page 212

Configuring Local Parameters (SRC CLI)

You create local parameters within a policy group. Use the following configuration statements to configure local parameters.

```
policies group name local-parameters name {
  description description ;
  default-value default-value ;
  type type ;
}
```

To configure local parameters:

1. From configuration mode, enter the local parameter configuration. For example, to configure a local parameter called bandwidthFactor:

```
user@host# edit policies group policer local-parameters bandwidthFactor
```

2. (Optional) Enter a description for the parameter. You can provide extra information and examples of how the parameter is used.

```
[edit policies group policer local-parameters bandwidthFactor]
user@host# set description description
```

3. (Optional) Configure a default value that the policy engine uses if no other values are provided during the parameter value acquisition process.

See “Parameter Types” on page 82 for valid values of each parameter type.

```
[edit policies group policer local-parameters bandwidthFactor]
user@host# set default-value default-value
```

4. (Optional) Set the type of attribute for which you can use the parameter.

```
[edit policies group policer local-parameters bandwidthFactor]
user@host# set type type
```

5. (Optional) Verify your configuration.

```
[edit policies group policer local-parameters bandwidthFactor]
user@host# show
default-value 1024*1024;
type rate;
```

- Related Topics**
- Overview of Global and Local Parameters on page 81
 - Configuring Local Parameters (C-Web Interface)
 - Configuring Global Parameters (SRC CLI) on page 101
 - Defining Parameters for the SRC Software on page 212

Viewing Runtime Parameters (SRC CLI)

Purpose Runtime parameters are parameters that are filled in with an actual value from the running system when the policy is installed. The SRC software comes with many predefined runtime parameters. Although the SRC CLI allows you to modify runtime parameters, you should not modify them. You can view a list of runtime parameters and view information about these parameters.

Action To view a list of runtime parameters:

```
user@host# edit policies global-parameters runtime-parameters
?
Possible completions:
  ah                Maps protocol 51 to AH
  any_priority      Sets packet loss priority to "any"
  any_protocol      Sets the drop profile protocol to "any"
  best_effort       Service flow scheduling type is best effort
  bootp            Specifies the BOOTP protocol
  . . .
  user_ipAddress    IP address of the subscriber
  user_ipMask       IP mask of the subscriber
  virtual_ipAddress Virtual portal address used with redundant SAEs
  winframe          Specifies the WinFrame protocol
  yellow            Sets the color of an action or classifier to yellow
```

To view information about runtime parameters:

```
user@host> show configuration policies global-parameters runtime-parameters
parameter interpolated {
  default-value "\"interpolate\"";
  type dropProfileType;
}
. . .
parameter low_priority {
  default-value "\"low\"";
  type packetLossPriority;
}
parameter ah {
  default-value 51;
  type protocol;
}
parameter sqlnet {
  default-value "\"sqlnet\"";
  type applicationProtocol;
}
parameter eq {
  default-value "\"eq\"";
  type portOperation;
}
```

- Related Topics**
- Configuring Global Parameters (SRC CLI) on page 101
 - Configuring Local Parameters (SRC CLI) on page 103
 - Viewing Runtime Parameters (C-Web Interface)
 - Viewing Predefined Global Parameters (SRC CLI) on page 101

Chapter 7

Configuring and Managing Policies (SRC CLI)

- Before You Configure SRC Policies on page 105
- Enabling the Policy Configuration on the SRC CLI on page 107
- Configuring Policy Folders (SRC CLI) on page 107
- Configuring Policy Groups (SRC CLI) on page 108
- Configuring Policy Lists (SRC CLI) on page 108
- Configuring Policy Rules on page 110
- Classify-Traffic Conditions on page 113
- Using Map Expressions in Application Protocol Conditions on page 147
- Configuring QoS Conditions (SRC CLI) on page 147
- Configuring Actions on page 148

Before You Configure SRC Policies

Building policies is a top-down operation. For example, before you can add a subordinate to the policy group, the policy group itself must exist.

Creating a Worksheet

Before you configure policies, you must determine what information you want to enter and where it will go. It is best to create a worksheet where you can record such things as names, priorities, addresses, and so on.

To create a worksheet:

1. Determine the policy requirements for your system.
2. Consider information that contains (at a minimum) names and parameters for:

- Policy group
- Policy list
- Policy rules
- Conditions
- Actions
- Record the policy information about the worksheet.

Naming Objects

Object names must be unique and must conform to LDAP distinguished name (DN) constraints.

Using the apply-groups Statement

When you use the **apply-groups** statement on the JUNOS routing platform to apply a configuration group to a hierarchy level in a configuration, you need to make sure that the SAE configuration group (default name is `sdx`) is in the first position in the **apply-groups** statement.

Using Expressions

Many of the policy options can take expressions in addition to literal values. If you can enter an expression for an option, the expression type is noted in the documentation for the command. For information about using and formatting expressions, see “Expressions in Parameters” on page 215.

Policy Values

As you are planning your policy configuration, you need to understand how invalid values in policies are handled on JUNOS routing platforms and JUNOSe routers.

SAE to JUNOS Routing Platforms

When the SAE sends policies to JUNOS routing platforms, it uses JUNOScript on the Blocks Extensible Exchange Protocol (BEEP), which is an XML-based protocol. Many of the configuration values in JUNOScript are strings in which the value is a number. If you enter an integer value that is too large, the policy software flags the entry as invalid, but the value is still sent to the router because JUNOScript on BEEP allows for its transmission. The router is the authority that decides whether values are valid for the particular version of the JUNOS software and the routing platform. If the value is too large, the router sends an error message to the SAE.

For example, the valid range for the burst size limit in a policer action is 1,500 to 100,000,000. If you specify a value greater than 100,000,000, it is flagged as invalid. However, as usual, the SRC software attempts to activate the service, but the activation will fail because the burst size is an invalid value on the router.

SAE to JUNOSe Routers

When the SAE sends policies to JUNOSe routers, it uses the Common Open Policy Service (COPS) protocol with specific standard Policy Information Bases (PIBs) and private PIBs. Many of the configuration values in the PIBs are not strings in which the value is a number. Sometimes the numeric range in the PIB is larger than the valid range of values on the router. For integer values in policies, the eventual policy on the router has only the portion of a value that can be converted to an integer in the usable range.

The example below for ToS byte is such a case. From the JUNOSe-IP-PIB:

```
JunoselpPolicyClacRuleEntry ::= SEQUENCE {
    junoselpPolicyClacRuleTosByte Integer32,
    junoselpPolicyClacRuleTosMask Integer32,
}
```

If a policy has a literal ToS byte value of 300, the high bits are ignored (or a mask of 255 is used) so that the value used for the ToS byte is 44; that is, 300 minus 256.

- Related Topics**
- Policy Management Overview on page 59
 - Overview of Policy Rules on page 110
 - Policy Information Model on page 62
 - Enabling the Policy Configuration on the SRC CLI on page 107

Enabling the Policy Configuration on the SRC CLI

Before you can configure policies with the SRC CLI, you must enable the policy, service, and subscriber editor on the SRC CLI. To do so:

In operational mode, enter the following command:

```
user@host> enable component editor
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Policy Components on page 60
 - Policy Information Model on page 62
 - Enabling the Policy Configuration on the C-Web Interface

Configuring Policy Folders (SRC CLI)

You use policy folders to organize policy groups. Use the following configuration statement to create a policy folder:

```
policies folder name ...
```

To create a policy folder:

- From configuration mode, enter the **edit policies folder** statement. For example, to create a folder called `junos_default`:

```
user@host# edit policies folder junos_default
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Enabling the Policy Configuration on the C-Web Interface
 - Example: Creating Access Policies for Subscribers on page 195

Configuring Policy Groups (SRC CLI)

Policy groups hold policy lists. You can create policy groups within policy folders. Use the following configuration statement to create a policy group:

```
policies group name {
    description description ;
}
```

To create a policy group:

1. From configuration mode, enter the **edit policies group** statement. For example, to create a folder called `dhcp-default`:

```
user@host# edit policies group dhcp-default
```

2. (Optional) Enter a description for the policy group.

```
[edit policies group dhcp-default]
user@host# set description description
```

3. (Optional) Verify your policy group configuration.

```
[edit policies group dhcp-default]
user@host#show
description "Default policy for JUNOSe routers";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Policy Folders (SRC CLI) on page 107
 - Enabling the Policy Configuration on the SRC CLI on page 107
 - Configuring Policy Groups (C-Web Interface)
 - Example: Creating Access Policies for Subscribers on page 195

Configuring Policy Lists (SRC CLI)

When you add a policy list, you specify whether the policy list is for JUNOS routing platforms (`junos`), JUNOSe routers (`junos-ip4`, `junos-ip6`), a CMTS device (`pcmm`),

or AAA devices (aaa). The type of policy list that you add controls the type of policy rules that you can add to the policy list.

You create policy lists within policy groups. Use the following configuration statements to create a policy list:

```
policies group name list name {
  role (junos | junose-ipv4 | junose-ipv6 | pcmm | aaa);
  applicability (input | output | both | secondary-input);
  description description ;
}
```

To add a policy list:

1. From configuration mode, create a policy list. For example, to create a policy list called *in* within a policy group called *dhcp*:

```
user@host# edit policies group dhcp list in
```

2. Specify the type of policy list. You must configure the type of policy list before you can add rules to the list.

```
[edit policies group dhcp list in]
user@host# set role junose-ipv4
```

3. Specify where the policy is applied on the router or, for PCMM policies, indicates whether the policy applies to the upstream or downstream channel. For JUNOS policies, you can specify whether a secondary policy applies to the input (ingress) side of the router interface.

```
[edit policies group dhcp list in]
user@host# set applicability input
```

4. (Optional) Provide a description of the policy list.

```
[edit policies group dhcp list in]
user@host# set description description
```

5. (Optional) Verify your policy list configuration.

```
[edit policies group dhcp list in]
user@host# show
role junose-ipv4;
applicability input;
description "input policy list for JUNOS DHCP";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Overview of Policy Rules on page 110
 - Enabling the Policy Configuration on the SRC CLI on page 107
 - Configuring Policy Lists (C-Web Interface)
 - Example: Creating Access Policies for Subscribers on page 195

Configuring Policy Rules

Policy rules specify traffic conditions and actions to be taken. Topics include:

- Overview of Policy Rules on page 110
- Adding a Policy Rule (SRC CLI) on page 111

Overview of Policy Rules

The type of policy rule that you can create depends on the type and applicability of the policy list in which you create the policy rule. There is only one type of policy rule for PCMM policy lists and AAA policy lists. For JUNOS policy lists, you can create JUNOS IPv4 or JUNOS IPv6 policy rule types. If you are creating a JUNOS secondary input policy, the applicability of policy list must be secondary-input. For JUNOS policy lists, you can create the following policy rule types:

- JUNOS ASP—Applicability of policy list must be both.
- JUNOS FILTER—Applicability of policy list must be input or output.
- JUNOS POLICER—Applicability of policy list must be input or output.
- JUNOS SCHEDULER—Applicability of policy list must be both.
- JUNOS SHAPING—Applicability of policy list must be both.

Before You Configure JUNOS Policy Rules

The following are prerequisites to using policy rules on JUNOS routing platforms:

- JUNOS Scheduler and JUNOS Shaping Policy Rules

Before you use the JUNOS scheduler and JUNOS shaping policy rules, check that your Physical Interface Card (PIC) supports JUNOS scheduling and shaping rate. Also, check that your interface supports the per-unit-scheduler.

You must enable the per-unit-scheduler on the interface. To do so, on the JUNOS routing platform, include the **per-unit-scheduler** statement at the [edit interfaces interface-name] hierarchy level:

```
[edit interfaces interface-name]
per-unit-scheduler;
```

- JUNOS ASP Policy Rules

Before you use the Adaptive Services PIC (ASP) policy rule to create a stateful firewall or NAT policy, you must configure the Adaptive Services PIC on the JUNOS routing platform. For example:

```
sp-0/1/0 {
  unit 0 {
    family inet {
      address 10.10.1.1/32;
    }
  }
}
```



```
}
```

For more information about configuring Adaptive Services PICs, see the *JUNOS Services Interfaces Configuration Guide*.

Setting the Policy Rule Precedence

Policy lists can have more than one policy rule. Policy rules are assigned a precedence that determines the order in which the policy manager applies policy rules. Rules are evaluated from lowest to highest precedence value. For JUNOSe policies, rules with equal precedence are evaluated in the order of creation. For JUNOS policies, rules with equal precedence are evaluated in random order.

Note that for JUNOS SCHEDULER and JUNOS POLICER policy rules, precedence is not a factor.

The router classifies packets beginning with the classify condition in the policy list that has the policy rule with the lowest precedence.

- If the packet matches the condition, the router applies the policy rule actions to the packet and does not continue to examine further conditions.
- If the packet does not match the condition, the router tries to match the packet with the classify condition in the policy rule with the next higher precedence.
- If the packet does not match any of the classify conditions, it is forwarded. There are some exceptions. For example, in the case of a JUNOS ASP stateful firewall, packets that do not match the classify conditions are dropped. Only matching packets are forwarded.

For JUNOSe routers, if you want the router to take two corresponding actions on a packet, you would create a JUNOSe policy list that has more than one policy rule with the same precedence. For example, you may want a policy rule that marks a packet and a policy rule that forwards the packet to the next interface. Or you could have a policy rule that applies a traffic class and a policy rule that forwards the packet to the next hop.

Related Topics

- Before You Configure SRC Policies on page 105
- Policy Information Model on page 62
- Adding a Policy Rule (SRC CLI) on page 111
- Enabling the Policy Configuration on the SRC CLI on page 107
- Example: Creating Access Policies for Subscribers on page 195

Adding a Policy Rule (SRC CLI)

You create policy rules within policy lists. Use the following configuration statements to create a policy rule:

```
policies group name list name rule name {
    type type ;
```

```

precedence precedence ;
accounting;
description description ;
}

```

To add a policy rule:

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 forward-dhcp within policy list input:

```

user@host# edit policies group dhcp list input rule forward-dhcp

```

2. Specify the type of policy rule.

The type of policy rule that you can create depends on the type and applicability of the policy list in which you create the policy rule.

```

[edit policies group dhcp list input rule forward-dhcp]
user@host# set type type

```

3. (Optional) Specify the order in which the policy manager applies rules.

```

[edit policies group dhcp list input rule forward-dhcp]
user@host# set precedence precedence

```

4. (Optional) Specify whether accounting data is collected for the actions specified in the rule.

```

[edit policies group dhcp list input rule forward-dhcp]
user@host# set accounting

```

5. (Optional) Provide a description of the policy rule.

```

[edit policies group dhcp list input rule forward-dhcp]
user@host# set description description

```

6. (Optional) Verify your policy rule configuration.

```

[edit policies group dhcp list input rule forward-dhcp]
user@host# show
type junose-ipv4;
precedence 200;
accounting;
description "Forward all dhcp packets from client to server";

```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Enabling the Policy Configuration on the SRC CLI on page 107
 - Overview of Policy Rules on page 110
 - Policy Information Model on page 62
 - Adding a Policy Rule

Classify-Traffic Conditions

Topics that discuss classify-traffic conditions include:

- Configuring Classify-Traffic Conditions on page 113
- Before You Configure Classify-Traffic Conditions on page 115
- Enabling Expansion of JUNOSe Classify-Traffic Conditions (SRC CLI) on page 115
- Specifying the PCMM Classifier Type (SRC CLI) on page 116
- Specifying Port Access for Traffic Classification (SRC CLI) on page 117
- Creating a Classify-Traffic Condition (SRC CLI) on page 118
- Configuring Source Networks (SRC CLI) on page 119
- Configuring Source Grouped Networks (SRC CLI) on page 120
- Configuring Destination Networks (SRC CLI) on page 121
- Configuring Destination Grouped Networks (SRC CLI) on page 122
- Configuring Protocol Conditions (SRC CLI) on page 123
- Configuring Protocol Conditions with Ports (SRC CLI) on page 124
- Configuring Protocol Conditions with Parameters (SRC CLI) on page 127
- Configuring TCP Conditions (SRC CLI) on page 131
- Configuring ICMP Conditions (SRC CLI) on page 134
- Configuring IGMP Conditions (SRC CLI) on page 135
- Configuring IPSec Conditions (SRC CLI) on page 136
- Configuring ToS Byte Conditions (SRC CLI) on page 138
- Configuring JUNOS Filter Conditions (SRC CLI) on page 139
- Configuring JUNOSe Parent-Group References (SRC CLI) on page 140
- Configuring JUNOSe Secondary Input Policy Conditions (SRC CLI) on page 142
- Configuring Application Protocol Conditions (SRC CLI) on page 144

Configuring Classify-Traffic Conditions

You create classify-traffic conditions in JUNOSe policy rules, in JUNOS ASP and JUNOS filter policy rules, and in PCMM policy rules.

The available configuration statements change depending on the type of policy rule that holds the condition and on the type of protocol that you specify.

To configure a classify-traffic condition, do the following:

1. Create a classify-traffic condition.
 - Creating a Classify-Traffic Condition (SRC CLI) on page 118
2. Configure source networks. You can configure source networks in one of two formats.
 - Configuring Source Networks (SRC CLI) on page 119

- Configuring Source Grouped Networks (SRC CLI) on page 120
- 3. Configure destination networks. You can configure destination networks in one of two formats.
 - Configuring Destination Networks (SRC CLI) on page 121
 - Configuring Destination Grouped Networks (SRC CLI) on page 122
- 4. Configure protocol conditions. The type of protocol condition that you use depends on your configuration.
 - To configure protocol conditions that do not include ports, see:
 - Configuring Protocol Conditions (SRC CLI) on page 123
 - To configure protocol conditions that include ports, see:
 - Configuring Protocol Conditions with Ports (SRC CLI) on page 124
 - To configure protocol conditions in which the protocol that you specify is a parameter, see:
 - Configuring Protocol Conditions with Parameters (SRC CLI) on page 127
 - To configure protocol conditions in which the protocol is TCP, see:
 - Configuring TCP Conditions (SRC CLI) on page 131
 - To configure protocol conditions in which the protocol is ICMP, see:
 - Configuring ICMP Conditions (SRC CLI) on page 134
 - To configure protocol conditions in which the protocol is IGMP, see:
 - Configuring IGMP Conditions (SRC CLI) on page 135
 - To configure protocol conditions in which the protocol is IPSec, see:
 - Configuring IPSec Conditions (SRC CLI) on page 136
 - To configure a ToS byte condition, see:
 - Configuring ToS Byte Conditions (SRC CLI) on page 138
- 5. For JUNOS filter policies, configure a JUNOS filter condition. See:
 - Configuring JUNOS Filter Conditions (SRC CLI) on page 139
- 6. For JUNOSe parent—group reference policies, configure a JUNOSe filter condition. See:

Configuring JUNOSe Parent-Group References (SRC CLI) on page 140
- 7. For JUNOSe secondary input policies, configure a traffic match condition for the packet flow. See:

- Configuring JUNOS Secondary Input Policy Conditions (SRC CLI) on page 142
8. For the stateful firewall and NAT policies, configure an application protocol condition. See:
- Configuring Application Protocol Conditions (SRC CLI) on page 144



NOTE: PCMM classifiers support only the following classifiers:

- Source and destination IP addresses
- Network protocol
- Source or destination port
- Type-of-service (ToS) byte and ToS mask

The policy engine ignores all other values.

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions (C-Web Interface)
 - Enabling Expansion of JUNOS Classify-Traffic Conditions (SRC CLI) on page 115
 - Specifying Port Access for Traffic Classification (SRC CLI) on page 117
 - Parameter Types on page 82

Before You Configure Classify-Traffic Conditions

If you are configuring classifiers for policies:

- For PCMM policies, you can specify that the classifier will be used in a PCMM IO2 or IO3 network. By default, the software translates classify-traffic conditions into PCMM IO2 classifiers.
- For JUNOS policies, you can specify that the SAE expand the classifier into multiple classifiers before it installs the policy on the router.

- Related Topics**
- Configuring Classify-Traffic Conditions on page 113
 - Enabling Expansion of JUNOS Classify-Traffic Conditions (SRC CLI) on page 115
 - Specifying the PCMM Classifier Type (SRC CLI) on page 116
 - Specifying Port Access for Traffic Classification (SRC CLI) on page 117
 - Creating a Classify-Traffic Condition (SRC CLI) on page 118

Enabling Expansion of JUNOS Classify-Traffic Conditions (SRC CLI)

For information about expanded classifiers, see “Policy Information Model” on page 62.

Use the following configuration statement to enable or disable the expansion of JUNOS classifiers.

```
shared sae configuration policy-management-configuration {
    enable-junose-classifier-expansion;
}
```

To enable or disable the expansion of JUNOS classifiers:

1. From configuration mode, access the configuration statement that configures policy management properties on the SAE.

```
user@host# edit shared sae configuration policy-management-configuration
```

2. Specify whether or not the SAE expands the JUNOS classify-traffic conditions into multiple classifiers before it installs the policy on the router.

```
[edit shared sae configuration policy-management-configuration]
user@host# set enable-junose-classifier-expansion
```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Specifying Port Access for Traffic Classification (SRC CLI) on page 117
 - Creating a Classify-Traffic Condition (SRC CLI) on page 118
 - Enabling Expansion of JUNOS Classify-Traffic Conditions (C-Web Interface)

Specifying the PCMM Classifier Type (SRC CLI)

Use the following configuration statement to specify which version of the PCMM classifiers you are using:

```
shared sae configuration driver pcmm {
    disable-pcmm-iO3-policy disable-pcmm-iO3-policy ;
}
```

To specify whether or not the SAE sends classifiers to the router that comply with PCMM IO3:

1. From configuration mode, access the configuration statement that configures the PCMM driver.

```
user@host# edit shared sae configuration driver pcmm
```

2. 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 configuration driver pcmm]
user@host# set disable-pcmm-iO3-policy disable-pcmm-iO3-policy
```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Delivering QoS Services in a Cable Environment on page 73
 - Configuring Classify-Traffic Conditions on page 113
 - Specifying the PCMM Classifier Type (C-Web Interface)

Specifying Port Access for Traffic Classification (SRC CLI)

In the SRC software, the way that you specify a range of port numbers greater than or less than a specific value in a traffic classifier is different from the way you define a range in the configuration on JUNOS routers.

In the SRC CLI, you specify ranges by setting values in the **port-operation** options in command statements.

To specify a range of port numbers greater or less than a specified value, you can:

- Define the full set of port numbers in the range to be allowed.
- Define the full set of port numbers in the range not allowed.

To configure port numbers greater than a defined value by specifying which values are allowed:

1. For the **port-operation** option, enter **eq**.
2. For the **from-port** option, enter the range of ports allowed.

For example, to specify access to all port numbers greater than 10, specify **11..65535**.

To configure port numbers greater than a defined value by specifying which values are not allowed:

1. For the **port-operation** option, enter **neq**.
2. For the **from-port** option, enter the range of ports not allowed.

For example, to specify access to all port numbers greater than 10, specify **1..9**.

To configure port numbers less than a defined value by specifying which values are allowed:

1. For the **port-operation** option, enter **eq**.
2. For the **from-port** option, enter the range of ports.

For example, to specify access to all port numbers less than 10, specify **1..9**.

To configure port numbers less than a defined value by specifying which values are not allowed:

1. For the **port-operation** option, enter **neq**.

2. For the **from-port** option, enter the range of ports.

For example, to specify access to all port numbers less than 10, specify **11..65535**.

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Enabling Expansion of JUNOS Classify-Traffic Conditions (SRC CLI) on page 115
 - Specifying the PCMM Classifier Type (SRC CLI) on page 116
 - Specifying Port Access for Traffic Classification (C-Web Interface)

Creating a Classify-Traffic Condition (SRC CLI)

You create classify-traffic conditions within policy rules. Use the following configuration statements to create a classify-traffic condition:

```
policies group name list name rule name traffic-condition name {
  match-direction match-direction ;
  description description ;
}
```

To add a classify-traffic condition:

1. From configuration mode, create a classify-traffic condition inside a policy rule that has already been created and configured. For example, to create a traffic-condition called **ctc** within policy rule **nat**:

```
user@host# edit policies group junos list staticnat rule nat traffic-condition ctc
```

2. (Optional) For JUNOS ASP policy rules, specify the direction of the packet flow on which you want to match packets.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc]
user@host# set match-direction match-direction
```

3. (Optional) Provide a description of the classify-traffic condition.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc]
user@host# set description description
```

4. (Optional) Verify your classify-traffic condition configuration.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc]
user@host# show
match-direction output;
description "Static NAT destination classifier";
```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113

- Enabling Expansion of JUNOSe Classify-Traffic Conditions (SRC CLI) on page 115
- Creating a Classify-Traffic Condition (C-Web Interface)
- Specifying Port Access for Traffic Classification (SRC CLI) on page 117

Configuring Source Networks (SRC CLI)

Use the following configuration statements to add source networks to a classify-traffic condition:

```

policies group name list name rule name traffic-condition name source-network
network {
  ip-address ip-address ;
  ip-mask ip-mask ;
  ip-operation ip-operation ;
}

```

To add a source network to a classify-traffic condition:

1. From configuration mode, enter the source network within a classify-traffic condition. For example:

```

user@host# edit policies group dhcp list in rule forward-dhcp traffic-condition
client-dhcp source-network network

```

2. (Optional) Configure the IP address of the source network or host.

```

[edit policies group dhcp list in rule forward-dhcp traffic-condition client-dhcp
source-network network]
user@host# set ip-address ip-address

```

3. (Optional) Configure the IP mask of the source network or host.

```

[edit policies group dhcp list in rule forward-dhcp traffic-condition client-dhcp
source-network network]
user@host# set ip-mask ip-mask

```

4. (Optional) Specify whether the software matches packets with an IP address that is equal or not equal to the specified address and mask.

```

[edit policies group dhcp list in rule forward-dhcp traffic-condition client-dhcp
source-network network]
user@host# set ip-operation ip-operation

```

5. (Optional) Verify your source network configuration.

```

[edit policies group dhcp list in rule forward-dhcp traffic-condition
client-dhcp source-network network]
user@host# show
ip-address interface_ipAddress;
ip-mask interface_ipMask;
ip-operation is_not;

```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring Source Grouped Networks (SRC CLI) on page 120
 - Configuring Source Networks (C-Web Interface)

Configuring Source Grouped Networks (SRC CLI)

You can configure source networks in grouped format. For JUNOS ASP and JUNOSe IPv6 policy rules, you must enter source networks in grouped format.

Use the following configuration statement to add source networks in a grouped format to a classify-traffic condition:

```
policies group name list name rule name traffic-condition name source-network
  group-network {
    network-specifier network-specifier ;
  }
```

To add a grouped source network to a classify-traffic condition:

1. From configuration mode, enter the source network within a classify-traffic condition. For example:

```
user@host# edit policies folder junose group dhcp list in rule forward-dhcp
traffic-condition client-dhcp source-network group-network
```

2. (Optional) Configure the IP address of the source network or host.

For JUNOS ASP policy rules, you must enter networks in the format `< ip address > / < prefix length >`. The `< ip address > / < mask >` format is rejected by the router.

For JUNOSe IPv6 policy rules, you must enter networks in the format `< ipv6 address > / < prefix length >`.

```
[edit policies folder junose group dhcp list in rule forward-dhcp traffic-condition
client-dhcp source-network group-network]
user@host# set network-specifier network-specifier
```

3. (Optional) Verify your source network configuration.

```
[edit policies folder junose group dhcp list in rule forward-dhcp
traffic-condition client-dhcp source-network group-network]
user@host# show
network-specifier gateway_ipAddress;
```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113

- Configuring Source Networks (SRC CLI) on page 119
- Configuring Source Grouped Networks (C-Web Interface)

Configuring Destination Networks (SRC CLI)

Use the following configuration statements to add destination networks to a classify-traffic condition:

```

policies group name list name rule name traffic-condition name
  destination-network network {
    ip-address ip-address ;
    ip-mask ip-mask ;
    ip-operation ip-operation ;
  }

```

To add a destination network to a classify-traffic condition:

1. From configuration mode, enter the destination network within a classify-traffic condition. For example:

```

user@host# edit policies group dhcp list in rule forward-dhcp traffic-condition
client-dhcp destination-network network

```

2. (Optional) Configure the IP address of the destination network or host.

```

[edit policies group dhcp list in rule forward-dhcp traffic-condition client-dhcp
destination-network network]
user@host# set ip-address ip-address

```

3. (Optional) Configure the IP mask of the destination network or host.

```

[edit policies group dhcp list in rule forward-dhcp traffic-condition client-dhcp
destination-network network]
user@host# set ip-mask ip-mask

```

4. (Optional) Specify whether the software matches packets with an IP address that is equal or not equal to the specified address and mask.

```

[edit policies group dhcp list in rule forward-dhcp traffic-condition client-dhcp
destination-network network]
user@host# set ip-operation ip-operation

```

5. (Optional) Verify your destination network configuration.

```

[edit policies group dhcp list in rule forward-dhcp traffic-condition
client-dhcp destination-network network]
user@host# show
ip-address interface_ipAddress;
ip-mask interfac_ipMask;
ip-operation is;

```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring Destination Grouped Networks (SRC CLI) on page 122
 - Configuring Destination Networks (C-Web Interface)

Configuring Destination Grouped Networks (SRC CLI)

You can configure destination networks in grouped format. For JUNOS ASP and JUNOS IPv6 policy rules, you must enter destination networks in grouped format.

Use the following configuration statements to add destination networks in a grouped format to a classify-traffic condition:

```
policies group name list name rule name traffic-condition name
  destination-network group-network {
    network-specifier network-specifier ;
  }
```

To add a grouped destination network to a classify-traffic condition:

1. From configuration mode, enter the destination network within a classify-traffic condition. For example:

```
user@host# edit policies group dhcp list in rule forward-dhcp traffic-condition
client-dhcp destination-network group-network
```

2. (Optional) Configure the IP address of the destination network or host.

For JUNOS ASP policy rules, you must enter networks in the format
< ip address > / < prefix length > .

For JUNOS IPv6 policy rules, you must enter networks in the format
< ipv6 address > / < prefix length > .

```
[edit policies group dhcp list in rule forward-dhcp traffic-condition client-dhcp
destination-network group-network]
user@host# set network-specifier network-specifier
```

3. (Optional) Verify your destination network configuration.

```
[edit policies group dhcp list in rule forward-dhcp traffic-condition
client-dhcp destination-network group-network]
user@host# show
network-specifier any;
```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring Destination Networks (SRC CLI) on page 121
 - Configuring Destination Grouped Networks (C-Web Interface)

Configuring Protocol Conditions (SRC CLI)

The procedure in this sections shows how to configure general protocol conditions.

- If your condition includes port numbers, use the procedure in “Configuring Protocol Conditions with Ports (SRC CLI)” on page 124.
- If your condition consists of a protocol that is assigned with a parameter value, use the procedure in “Configuring Protocol Conditions with Parameters (SRC CLI)” on page 127.

Use the following configuration statements to add general protocol conditions to a classify-traffic condition:

```

policies group name list name rule name traffic-condition name protocol-condition
{
  protocol protocol ;
  protocol-operation protocol-operation ;
  ip-flags ip-flags ;
  ip-flags-mask ip-flags-mask ;
  fragment-offset fragment-offset ;
  packet-length packet-length ;
}

```

To add general protocol conditions to a classify-traffic condition:

1. From configuration mode, enter the general protocol condition configuration. For example:

```

user@host# edit policies group dhcp list in rule forward-dhcp traffic-condition
client-dhcp protocol-condition

```

2. Configure the protocol matched by this classify-traffic condition.

```

[edit policies group dhcp list in rule forward-dhcp traffic-condition client-dhcp
protocol-condition]
user@host# set protocol protocol

```

3. Configure the policy to match packets with the protocol that is either equal or not equal to the specified protocol.

```

[edit policies group dhcp list in rule forward-dhcp traffic-condition client-dhcp
protocol-condition]
user@host# set protocol-operation protocol-operation

```

4. (Optional) Configure the value of the IP flags field in the IP header.

```

[edit policies group dhcp list in rule forward-dhcp traffic-condition client-dhcp
protocol-condition]
user@host# set ip-flags ip-flags

```

5. (Optional) Configure the mask that is associated with the IP flag.

```

[edit policies group dhcp list in rule forward-dhcp traffic-condition client-dhcp
protocol-condition]

```

```
user@host# set ip-flags-mask ip-flags-mask
```

6. (Optional) Configure the value of the fragment offset field.

```
[edit policies group dhcp list in rule forward-dhcp traffic-condition client-dhcp
protocol-condition]
```

```
user@host# set fragment-offset fragment-offset
```

7. (Optional) Configure the packet length on which to match. The length refers only to the IP packet, including the packet header, and does not include any layer 2 encapsulation overhead.

```
[edit policies group dhcp list in rule forward-dhcp traffic-condition client-dhcp
protocol-condition]
```

```
user@host# set packet-length packet-length
```

8. (Optional) Verify your protocol condition configuration.

```
[edit policies group dhcp list in rule forward-dhcp traffic-condition
client-dhcp protocol-condition]
```

```
user@host# show
protocol 0;
protocol-operation 1;
ip-flags 0;
ip-flags-mask 0;
fragment-offset any;
```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring Protocol Conditions (C-Web Interface)

Configuring Protocol Conditions with Ports (SRC CLI)

Use the following configuration statements to add general protocol conditions with ports to a classify-traffic condition:

```
policies group name list name rule name traffic-condition name
  protocol-port-condition {
    protocol protocol ;
    protocol-operation protocol-operation ;
    ip-flags ip-flags ;
    ip-flags-mask ip-flags-mask ;
    fragment-offset fragment-offset;
    packet-length packet-length ;
  }
policies group name list name rule name traffic-condition name
  protocol-port-condition destination-port port {
    port-operation port-operation ;
    from-port from-port ;
  }
```

```

policies group name list name rule name traffic-condition name
  protocol-port-condition source-port port {
    port-operation port-operation ;
    from-port from-port ;
  }

```

To add general protocol conditions with ports to a classify-traffic condition:

1. From configuration mode, enter the protocol port condition configuration. For example:

```

user@host# edit policies group junos list bodVpn rule pr traffic-condition ctc
protocol-port-condition

```

2. Configure the protocol matched by this classify-traffic condition.

```

[edit policies group junos list bodVpn rule pr traffic-condition ctc
 protocol-port-condition]
user@host# set protocol protocol

```

3. Configure the policy to match packets with the protocol that is either equal or not equal to the specified protocol.

```

[edit policies group junos list bodVpn rule pr traffic-condition ctc
 protocol-port-condition]
user@host# set protocol-operation protocol-operation

```

4. (Optional) Configure the value of the IP flags field in the IP header.

```

[edit policies group junos list bodVpn rule pr traffic-condition ctc
 protocol-port-condition]
user@host# set ip-flags ip-flags

```

5. (Optional) Configure the mask that is associated with the IP flag.

```

[edit policies group junos list bodVpn rule pr traffic-condition ctc
 protocol-port-condition]
user@host# set ip-flags-mask ip-flags-mask

```

6. (Optional) Configure the value of the fragment offset field.

```

[edit policies group junos list bodVpn rule pr traffic-condition ctc
 protocol-port-condition]
user@host# set fragment-offset fragment-offset

```

7. (Optional) Configure the packet length on which to match. The length refers only to the IP packet, including the packet header, and does not include any layer 2 encapsulation overhead.

```

[edit policies group junos list bodVpn rule pr traffic-condition ctc
 protocol-port-condition]
user@host# set packet-length packet-length

```

8. (Optional) Enter the destination port configuration for the protocol port configuration.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
protocol-port-condition]
user@host# edit destination-port
```

9. (Optional) Configure the policy to match packets with a port that is either equal or not equal to the specified port.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
protocol-port-condition destination-port port]
user@host# set port-operation port-operation
```

10. (Optional) Configure the destination port.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
protocol-port-condition destination-port port]
user@host# set from-port from-port
```

11. (Optional) Enter the source port configuration for the protocol port configuration.

```
user@host# up
```

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
protocol-port-condition]
user@host# edit source-port
```

12. (Optional) Configure the policy to match packets with a port that is either equal or not equal to the specified port.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
protocol-port-condition source-port port]
user@host# set port-operation port-operation
```

13. (Optional) Configure the source port.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
protocol-port-condition source-port port]
user@host# set from-port from-port
```

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
protocol-port-condition source-port port]
user@host# up
```

14. (Optional) Verify your protocol condition configuration.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
protocol-port-condition]
user@host# show
protocol 17;
protocol-operation 1;
ip-flags ipFlags;
```



```

ip-flags-mask ipFlagsMask;
fragment-offset ipFragOffset;
packet-length packetLength;
destination-port {
    port {
        port-operation eq;
        from-port service_port;
    }
}
source-port {
    port {
        port-operation eq;
        from-port service_port;
    }
}

```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring Protocol Conditions with Ports (C-Web Interface)
 - Configuring Protocol Conditions (SRC CLI) on page 123
 - Configuring Protocol Conditions with Parameters (SRC CLI) on page 127

Configuring Protocol Conditions with Parameters (SRC CLI)

Use the following configuration statements to configure classify-traffic conditions that contain a parameter value for the protocol:

```

policies group name list name rule name traffic-condition name
    parameter-protocol-condition {
        protocol protocol ;
        protocol-operation protocol-operation ;
        tcp-flags tcp-flags ;
        tcp-flags-mask tcp-flags-mask ;
        spi spi ;
        ip-flags ip-flags ;
        ip-flags-mask ip-flags-mask ;
        fragment-offset fragment-offset ;
        packet-length packet-length ;
    }
policies group name list name rule name traffic-condition name
    parameter-protocol-condition proto-attr {
        icmp-type icmp-type ;
        icmp-code icmp-code ;
        igmp-type igmp-type ;
    }
policies group name list name rule name traffic-condition name
    parameter-protocol-condition proto-attr destination-port port {
        port-operation port-operation ;
        from-port from-port ;
    }
policies group name list name rule name traffic-condition name
    parameter-protocol-condition proto-attr source-port port {

```

```

port-operation port-operation ;
from-port from-port ;
}

```

To configure a protocol condition that contains a parameter value for the protocol:

1. From configuration mode, enter the parameter protocol condition configuration. For example:

```

user@host# edit policies group junose list dhcp rule forward-dhcp
traffic-condition ctc parameter-protocol-condition

```

2. Assign a parameter as the protocol matched by this classify-traffic condition.

Before you assign a parameter, you must create a parameter of type protocol and commit the parameter configuration.

```

[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition]
user@host# set protocol protocol

```

3. (Optional) Configure the policy to match packets with the protocol that is either equal or not equal to the specified protocol.

```

[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition]
user@host# set protocol-operation protocol-operation

```

4. (Optional) Configure the value of the TCP flags field in the IP header.

```

[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition]
user@host# set tcp-flags tcp-flags

```

5. (Optional) Configure the mask associated with TCP flags.

```

[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition]
user@host# set tcp-flags-mask tcp-flags-mask

```

6. (Optional) Specify the authentication header (AH) or the encapsulating security payload (ESP) security parameter index (SPI).

```

[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition]
user@host# set spi spi

```

7. (Optional) Configure the value of the IP flags field in the IP header.

```

[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition]
user@host# set ip-flags ip-flags

```

8. (Optional) Configure the mask that is associated with the IP flag.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition]
user@host# set ip-flags-mask ip-flags-mask
```

9. (Optional) Configure the value of the fragment offset field.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition]
user@host# set fragment-offset fragment-offset
```

10. (Optional) Configure the packet length on which to match. The length refers only to the IP packet, including the packet header, and does not include any layer 2 encapsulation overhead.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition]
user@host# set packet-length packet-length
```

11. (Optional) Enter the protocol attribute configuration.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition]
user@host# edit proto-attr
```

12. (Optional) Configure the ICMP packet type.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition proto-attr]
user@host# set icmp-type icmp-type
```

13. (Optional) Configure the ICMP code.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition proto-attr]
user@host# set icmp-code icmp-code
```

14. (Optional) Configure the IGMP packet type on which to match.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition proto-attr]
user@host# set igmp-type igmp-type
```

15. (Optional) Enter the destination port configuration.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition proto-attr]
user@host# edit destination-port port
```

16. (Optional) Configure the policy to match packets with a port that is either equal or not equal to the specified port.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition proto-attr destination-port port]
```

```
user@host# set port-operation port-operation
```

17. (Optional) Configure the TCP or UDP destination port.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition proto-attr destination-port port]
user@host# set from-port from-port
```

18. (Optional) Enter the source port configuration.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition proto-attr destination-port port]
user@host# up
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
param-proto-condition proto-attr]
user@host# edit source-port port
```

19. (Optional) Configure the policy to match packets with a port that is either equal or not equal to the specified port.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition proto-attr source-port port]
user@host# set port-operation port-operation
```

20. (Optional) Configure the TCP or UDP source port.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition proto-attr source-port port]
user@host# set from-port from-port
```

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition proto-attr source-port port]
user@host# up
```

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition proto-attr source-port]
user@host# up
```

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition proto-attr]
user@host# up
```

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition ctc
parameter-protocol-condition]
user@host# up
```

21. (Optional) Verify the parameter protocol configuration.

```
[edit policies group junose list dhcp rule forward-dhcp traffic-condition
ctc parameter-protocol-condition]
user@host# show
protocol protocol;
```

```

protocol-operation is;
tcp-flags 0;
tcp-flags-mask 0;
ip-flags 0;
ip-flags-mask 0;
proto-attr {
    icmp-type 255;
    icmp-code 255;
    destination-port {
        port {
            port-operation eq;
            from-port outsidePort;
        }
    }
}

```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring Protocol Conditions with Parameters (C-Web Interface)
 - Configuring Protocol Conditions (SRC CLI) on page 123

Configuring TCP Conditions (SRC CLI)

Use the following configuration statements to add TCP conditions to a classify-traffic condition:

```

policies group name list name rule name traffic-condition name tcp-condition {
    tcp-flags tcp-flags ;
    tcp-flags-mask tcp-flags-mask ;
    protocol protocol ;
    protocol-operation protocol-operation ;
    ip-flags ip-flags ;
    ip-flags-mask ip-flags-mask ;
    fragment-offset fragment-offset ;
    packet-length packet-length ;
}

```

Because the protocol is already set to TCP, do not change the protocol or protocol-operation options.

```

policies group name list name rule name traffic-condition name tcp-condition
    destination-port port {
        port-operation port-operation ;
        from-port from-port ;
    }
policies group name list name rule name traffic-condition name tcp-condition
    source-port port {
        port-operation port-operation ;
        from-port from-port ;
    }

```

To add TCP conditions to a classify-traffic condition:

1. From configuration mode, enter the TCP configuration. For example:

```
user@host# edit policies group junos list tcpCondition rule pr traffic-condition  
ctc tcp-condition
```

2. (Optional) Configure the value of the TCP flags field in the IP header.

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition]  
user@host# set tcp-flags tcp-flags
```

3. (Optional) Configure the mask associated with TCP flags.

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition]  
user@host# set tcp-flags-mask tcp-flags-mask
```

4. (Optional) Configure the value of the IP flags field in the IP header.

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition]  
user@host# set ip-flags ip-flags
```

5. (Optional) Configure the mask that is associated with the IP flag.

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition]  
user@host# set ip-flags-mask ip-flags-mask
```

6. (Optional) Configure the value of the fragment offset field.

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition]  
user@host# set fragment-offset fragment-offset
```

7. (Optional) For JUNOS filter policies, configure the packet length on which to match. The length refers only to the IP packet, including the packet header, and does not include any layer 2 encapsulation overhead.

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition]  
user@host# set packet-length packet-length
```

8. (Optional) Enter the destination port configuration for the TCP configuration.

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition]  
user@host# edit destination-port port
```

9. (Optional) Configure the policy to match packets with a port that is either equal or not equal to the specified port.

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition  
destination-port port]  
user@host# set port-operation port-operation
```

10. (Optional) Configure the destination port.

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition  
destination-port port]
```

```
user@host# set from-port from-port
```

11. (Optional) Enter the source port configuration for the TCP configuration.

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition
destination-port port]
```

```
user@host# up
```

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition]
```

```
user@host# edit source-port port
```

12. (Optional) Configure the policy to match packets with a port that is either equal or not equal to the specified port.

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition
source-port port]
```

```
user@host# set port-operation port-operation
```

13. (Optional) Configure the source port.

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition
source-port port]
```

```
user@host# set from-port from-port
```

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition
source-port port]
```

```
user@host# up
```

```
[edit policies group junos list tcpCondition rule pr traffic-condition ctc tcp-condition
source-port]
```

```
user@host# up
```

14. (Optional) Verify the TCP condition configuration.

```
[edit policies group junos list tcpCondition rule pr traffic-condition
ctc tcp-condition]
```

```
user@host# show
```

```
tcp-flags 0;
tcp-flags-mask 0;
protocol tcp;
protocol-operation is;
ip-flags 0;
ip-flags-mask 0;
destination-port {
  port {
    port-operation eq;
    from-port service_port;
  }
}
source-port {
  port {
    port-operation eq;
    from-port service_port;
  }
}
```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring TCP Conditions (C-Web Interface)

Configuring ICMP Conditions (SRC CLI)

Use the following configuration statements to add ICMP conditions to a classify-traffic condition:

```

policies group name list name rule name traffic-condition name icmp-condition
{
  protocol protocol ;
  protocol-operation protocol-operation ;
  ip-flags ip-flags ;
  ip-flags-mask ip-flags-mask ;
  fragment-offset fragment-offset ;
  packet-length packet-length ;
  icmp-type icmp-type ;
  icmp-code icmp-code ;
}

```

Because the protocol is already set to ICMP, do not change the protocol or protocol-operation options.

To add ICMP conditions to a classify-traffic condition:

1. From configuration mode, enter the ICMP configuration. For example:

```

user@host# edit policies group bod list input rule pr traffic-condition ctc
icmp-condition

```

2. (Optional) Configure the value of the IP flags field in the IP header.

```

[edit policies group bod list input rule pr traffic-condition ctc icmp-condition]
user@host# set ip-flags ip-flags

```

3. (Optional) Configure the mask that is associated with the IP flag.

```

[edit policies group bod list input rule pr traffic-condition ctc icmp-condition]
user@host# set ip-flags-mask ip-flags-mask

```

4. (Optional) Configure the value of the fragment offset field.

```

[edit policies group bod list input rule pr traffic-condition ctc icmp-condition]
user@host# set fragment-offset fragment-offset

```

5. (Optional) Configure the packet length on which to match. The length refers only to the IP packet, including the packet header, and does not include any layer 2 encapsulation overhead.

```

[edit policies group bod list input rule pr traffic-condition ctc icmp-condition]
user@host# set packet-length packet-length

```


6. (Optional) Configure the ICMP packet type on which to match. The packet type must be supported by the router or CMTS device.

```
[edit policies group bod list input rule pr traffic-condition ctc icmp-condition]
user@host# set icmp-type icmp-type
```

7. (Optional) Configure the ICMP code on which to match. The ICMP code must be supported by the router or CMTS device.

```
[edit policies group bod list input rule pr traffic-condition ctc icmp-condition]
user@host# set icmp-code icmp-code
```

8. (Optional) Verify the ICMP condition configuration.

```
[edit policies group bod list input rule pr traffic-condition ctc
icmp-condition]
user@host# show
protocol icmp;
protocol-operation 1;
ip-flags ipFlags;
ip-flags-mask ipFlagsMask;
fragment-offset ipFragOffset;
icmp-type icmpType;
icmp-code icmpCode;
```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring ICMP Conditions (C-Web Interface)

Configuring IGMP Conditions (SRC CLI)

Use the following configuration statements to add IGMP conditions to a classify-traffic condition:

```
policies group name list name rule name traffic-condition name igmp-condition
{
  protocol protocol ;
  protocol-operation protocol-operation ;
  ip-flags ip-flags;
  ip-flags-mask ip-flags-mask ;
  fragment-offset fragment-offset ;
  packet-length packet-length ;
  igmp-type igmp-type ;
}
```

Because the protocol is already set to IGMP, do not change the protocol or protocol-operation options.

To add IGMP conditions to a classify-traffic condition:

1. From configuration mode, enter the IGMP configuration. For example:

```
user@host# edit policies group junose list pl rule pr traffic-condition ctc igmp-condition
```

2. (Optional) Configure the value of the IP flags field in the IP header.

```
[edit policies group junose list pl rule pr traffic-condition ctc igmp-condition]
user@host# set ip-flags ip-flags
```

3. (Optional) Configure the mask that is associated with the IP flag.

```
[edit policies group junose list pl rule pr traffic-condition ctc igmp-condition]
user@host# set ip-flags-mask ip-flags-mask
```

4. (Optional) Configure the value of the fragment offset field.

```
[edit policies group junose list pl rule pr traffic-condition ctc igmp-condition]
user@host# set fragment-offset fragment-offset
```

5. (Optional) Configure the packet length on which to match. The length refers only to the IP packet, including the packet header, and does not include any layer 2 encapsulation overhead.

```
[edit policies group junose list pl rule pr traffic-condition ctc igmp-condition]
user@host# set packet-length packet-length
```

6. (Optional) Configure the IGMP packet type on which to match.

```
[edit policies group junose list pl rule pr traffic-condition ctc igmp-condition]
user@host# set igmp-type icmp-type
```

7. (Optional) Verify the IGMP condition configuration.

```
[edit policies group junose list pl rule pr traffic-condition ctc igmp-condition]
user@host# show
protocol igmp;
protocol-operation 1;
ip-flags 0;
ip-flags-mask 0;
fragment-offset 0;
igmp-type igmpType;
```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring IGMP Conditions (C-Web Interface)

Configuring IPSec Conditions (SRC CLI)

You can configure IPSec conditions for JUNOS policy rules. Use the following configuration statements to add IPSec conditions to a classify-traffic condition:

```

policies group name list name rule name traffic-condition name ipsec-condition
{
    spi spi ;
    ip-flags ip-flags ;
    ip-flags-mask ip-flags-mask ;
    fragment-offset fragment-offset ;
    packet-length packet-length ;
    protocol protocol ;
    protocol-operation protocol-operation;
}

```

To add IPsec conditions to a classify-traffic condition:

1. From configuration mode, enter the IPsec configuration. For example:

```

user@host# edit policies group vpn list input rule pr traffic-condition ctc
ipsec-condition

```

2. (Optional) Specify the authentication header (AH) or the encapsulating security payload (ESP) security parameter index (SPI).

```

[edit policies group vpn list input rule pr traffic-condition ctc ipsec-condition]
user@host# set spi spi

```

3. (Optional) Configure the value of the IP flags field in the IP header.

```

[edit policies group vpn list input rule pr traffic-condition ctc ipsec-condition]
user@host# set ip-flags ip-flags

```

4. (Optional) Configure the mask that is associated with the IP flag.

```

[edit policies group vpn list input rule pr traffic-condition ctc ipsec-condition]
user@host# set ip-flags-mask ip-flags-mask

```

5. (Optional) Configure the value of the fragment offset field.

```

[edit policies group vpn list input rule pr traffic-condition ctc ipsec-condition]
user@host# set fragment-offset fragment-offset

```

6. (Optional) Configure the packet length on which to match. The length refers only to the IP packet, including the packet header, and does not include any layer 2 encapsulation overhead.

```

[edit policies group vpn list input rule pr traffic-condition ctc ipsec-condition]
user@host# set packet-length packet-length

```

7. Configure the protocol matched by this classify-traffic condition.

```

[edit policies group vpn list input rule pr traffic-condition ctc ipsec-condition]
user@host# set protocol protocol

```

8. (Optional) Verify the IPsec condition configuration.

```
[edit policies group vpn list input rule pr traffic-condition ctc
ipsec-condition
user@host# show
spi 2;
ip-flags 0;
ip-flags-mask 0;
fragment-offset 0;
packet-length packetLength;
protocol ah;
protocol-operation 1;
```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring IPSec Conditions (C-Web Interface)

Configuring ToS Byte Conditions (SRC CLI)

Use this condition to define a particular traffic flow to the service's network for the DA IP field in the IP packet.

The CoS feature on JUNOS routing platforms supports DiffServ as well as six-bit IP header ToS byte settings. The DiffServ protocol uses the ToS byte in the IP header. The most significant six bits of this byte form the Differentiated Services code point (DSCP). The CoS feature uses DSCPs to determine the forwarding class associated with each packet. It also uses the ToS byte and ToS byte mask to determine IP precedence.

Use the following configuration statements to add ToS conditions to a classify-traffic condition:

```
policies group name list name rule name traffic-condition name tos {
  tos-byte tos-byte ;
  tos-byte-mask tos-byte-mask ;
}
```

To add ToS conditions to a classify-traffic condition:

1. From configuration mode, enter the ToS configuration. For example:

```
user@host# edit policies group junos list bodVpn rule pr traffic-condition ctc
tos
```

2. (Optional) Configure the value of the ToS byte in the IP packet header.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc tos]
user@host# set tos-byte tos-byte
```

3. (Optional) Configure the mask associated with the ToS byte.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc tos]
user@host# set tos-byte-mask tos-byte-mask
```

4. (Optional) Verify the ToS condition configuration.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc tos]
user@host# show
tos-byte tosByte;
tos-byte-mask tosMask;
```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113

Configuring JUNOS Filter Conditions (SRC CLI)

Use the following configuration statements to configure JUNOS filter conditions.

```
policies group name list name rule name traffic-condition name
  traffic-match-condition {
    forwarding-class forwarding-class ;
    interface-group interface-group ;
    source-class source-class ;
    destination-class destination-class ;
    allow-ip-options allow-ip-options ;
  }
```

To add JUNOS filter conditions to a classify-traffic condition:

1. From configuration mode, enter the application protocol configuration. For example:

```
user@host# edit policies group junos list bodVpn rule pr traffic-condition ctc
traffic-match-condition
```

2. (Optional) Configure the name of a forwarding class to match.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
traffic-match-condition]
user@host# set forwarding-class forwarding-class
```

3. (Optional) Configure the condition to match packets based on the interface group on which the packet was received.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
traffic-match-condition]
user@host# set interface-group interface-group
```

4. (Optional) Configure the condition to match packets based on source class. A source class is a set of source prefixes grouped together and given a class name. You usually match source and destination classes for output firewall filters.

You cannot match on both source class and destination class at the same time. You must choose one or the other.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
traffic-match-condition]
user@host# set source-class source-class
```

5. (Optional) Configure the condition to match packets based on destination class. A destination class is a set of destination prefixes grouped together and given a class name. You usually match source and destination classes for output firewall filters.

You cannot match on both source class and destination class at the same time. You must choose one or the other.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
traffic-match-condition]
user@host# set destination-class destination-class
```

6. (Optional) Configure the condition to match packets based on IP options.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
traffic-match-condition]
user@host# set allow-ip-options allow-ip-options
```

7. (Optional) Verify the JUNOS filter condition configuration.

```
[edit policies group junos list bodVpn rule pr traffic-condition ctc
traffic-match-condition]
user@host# show
forwarding-class fc_expedited;
interface-group 42;
source-class gold-class;
destination-class gold-class;
allow-ip-options strict-source-route;
```

- Related Topics**
- Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring JUNOS Filter Conditions (C-Web Interface)

Configuring JUNOS Parent-Group References (SRC CLI)

For JUNOS policies, you can apply filter conditions to the input (ingress) and output (egress) side of the router interface. Typically, you use filter conditions with hierarchical rate-limit actions.

Use the following configuration statements to configure match conditions for a JUNOS filter

```
policies group name list name rule name traffic-condition name
traffic-match-condition {
parent-group parent-group
source-class source-class ;
destination-class destination-class ;
```

```

traffic-class traffic-class;
color color;
user-packet-class user-packet-class;
destination-local-interface destination-local-interface ;
}

```

To add JUNOS filter conditions to a classify-traffic condition:

1. From configuration mode, create a classify-traffic condition. For example, to create a traffic-condition called `tcpg` within policy rule `pr`:

```

[edit]
user@host# edit policies group junose list ipv4 rule pr traffic-condition tcpg
traffic-match-condition

```

2. (Optional) Configure the condition to match one match conditions.

```

[edit policies group junose list ipv4 rule pr traffic-condition tcpg
traffic-match-condition]
user@host# set parent-group parent-group

```

3. (Optional) Configure the condition to match one or more parent groups in a hierarchical rate-limit action.

```

[edit policies group junose list ipv4 rule pr traffic-condition tcpg
traffic-match-condition]
user@host# set parent-group parent-group

```

4. (Optional) Configure the condition to match packets based on the source route class.

```

[edit policies group junose list ipv4 rule pr traffic-condition tcpg
traffic-match-condition]
user@host# set source-class source-class

```

5. (Optional) Configure the condition to match packets based on the destination route class.

```

[edit policies group junose list ipv4 rule pr traffic-condition tcpg
traffic-match-condition]
user@host# set destination-class destination-class

```

6. (Optional) Configure the condition to match packets based on the traffic class.

```

[edit policies group junose list ipv4 rule pr traffic-condition tcpg
traffic-match-condition]
user@host# set traffic-class traffic-class

```

7. (Optional) Configure the condition to match packets based on the packet color.

```

[edit policies group junose list ipv4 rule pr traffic-condition tcpg
traffic-match-condition]
user@host# set color color

```

8. (Optional) Configure the condition to match packets based on the user packet class action number.

```
[edit policies group junose list ipv4 rule pr traffic-condition tcpg
traffic-match-condition]
user@host# set user-packet-class user-packet-class
```

9. (Optional) Configure the condition to match packets based on packets destined for a local interface.

```
[edit policies group junose list ipv4 rule pr traffic-condition tcpg
traffic-match-condition]
user@host# set destination-local-interface destination-local-interface
```

10. (Optional) Verify the secondary input policy configuration.

```
[edit policies group junose list ipv4 rule pr traffic-condition tcpg
traffic-match-condition]
user@host# show
```

- Related Topics**
- Configuring Policy Lists (SRC CLI) on page 108
 - Before You Configure Classify-Traffic Conditions on page 115
 - Overview of Policy Rules on page 110
 - Configuring Classify-Traffic Conditions on page 113
 - Dynamic Bandwidth Allocation Among Traffic Flows in SRC Policy Lists for JUNOS Routing Platforms on page 71

Configuring JUNOS Secondary Input Policy Conditions (SRC CLI)

For JUNOS policies, you can apply secondary input policies to the input (ingress) side of the router interface. Secondary input policies evaluate conditions after a route lookup.

Use the following configuration statements to configure match conditions for JUNOS secondary input policies:

```
policies group name list name rule name traffic-condition name
  traffic-match-condition {
    source-class source-class ;
    destination-class destination-class ;
    traffic-class traffic-class;
    color color;
    user-packet-class user-packet-class;
    destination-local-interface destination-local-interface ;
  }
```

To add conditions for JUNOS secondary input policies to a classify-traffic condition:

1. From configuration mode, create a classify-traffic condition inside a policy rule that has already been created and configured for a policy list whose type is `junose-ipv4` or `junose-ipv6` and applicability is `secondary-input`. For example, to create a traffic-condition called `rtcl` within policy rule `clacl`:


```
user@host# edit policies group junose list ipv4 rule clacl traffic-condition rtcl  
traffic-match-condition
```

2. (Optional) Configure the condition to match packets based on the source route class.

```
[edit policies group junose list ipv4 rule clacl traffic-condition rtcl  
traffic-match-condition]  
user@host# set source-class source-class
```

3. (Optional) Configure the condition to match packets based on the destination route class.

```
[edit policies group junose list ipv4 rule clacl traffic-condition rtcl  
traffic-match-condition]  
user@host# set destination-class destination-class
```

4. (Optional) Configure the condition to match packets based on the traffic class.

```
[edit policies group junose list ipv4 rule clacl traffic-condition rtcl  
traffic-match-condition]  
user@host# set traffic-class traffic-class
```

5. (Optional) Configure the condition to match packets based on the packet color.

```
[edit policies group junose list ipv4 rule clacl traffic-condition rtcl  
traffic-match-condition]  
user@host# set color color
```

6. (Optional) Configure the condition to match packets based on the user packet class action number.

```
[edit policies group junose list ipv4 rule clacl traffic-condition rtcl  
traffic-match-condition]  
user@host# set user-packet-class user-packet-class
```

7. (Optional) Configure the condition to match packets based on packets destined for a local interface.

```
[edit policies group junose list ipv4 rule clacl traffic-condition rtcl  
traffic-match-condition]  
user@host# set destination-local-interface destination-local-interface
```

8. (Optional) Verify the secondary input policy configuration.

```
[edit policies group junose list ipv4 rule clacl traffic-condition rtcl  
traffic-match-condition]  
user@host# show
```

- Related Topics**
- Configuring Policy Lists (SRC CLI) on page 108
 - Before You Configure Classify-Traffic Conditions on page 115
 - Overview of Policy Rules on page 110
 - Configuring Classify-Traffic Conditions on page 113

Configuring Application Protocol Conditions (SRC CLI)

You can define application protocols for the stateful firewall and NAT services to use in match condition rules. An application protocol defines application parameters by using information from network layer 3 and above. Examples of such applications are FTP and H.323.

Use the following configuration statements to add application protocol conditions to a classify-traffic condition:

```

policies group name list name rule name traffic-condition name
  application-protocol-condition name {
    protocol protocol ;
    application-protocol application-protocol ;
    idle-timeout idle-timeout ;
    dce-rpc-uuid dce-rpc-uuid ;
    rpc-program-number rpc-program-number ;
    snmp-command snmp-command ;
    ttl-threshold ttl-threshold ;
  }
policies group name list name rule name traffic-condition name
  application-protocol-condition name proto-attr {
    icmp-type icmp-type ;
    icmp-code icmp-code ;
  }
policies group name list name rule name traffic-condition name
  application-protocol-condition name proto-attr destination-port port {
    from-port from-port ;
  }
policies group name list name rule name traffic-condition name
  application-protocol-condition name proto-attr source-port port {
    from-port from-port ;
  }

```

To add application protocol conditions to a classify-traffic condition:

1. From configuration mode, enter the application protocol configuration. In this procedure, *apc* is the name of the application protocol condition. For example:

```

user@host# edit policies group junos list staticnat rule nat traffic-condition ctc
application-protocol-condition apc

```

2. (Optional) Configure the network protocol to match.

```

[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc]
user@host# set protocol protocol

```

3. (Optional) Configure the application protocol to match.

```

[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc]
user@host# set application-protocol application-protocol

```

4. (Optional) Configure the length of time the application is inactive before it times out.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc]
user@host# set idle-timeout idle-timeout
```

5. (Optional) For the DCE RPC application protocol, configure the universal unique identifier (UUID).

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc]
user@host# set dce-rpc-uuid dce-rpc-uuid
```

6. (Optional) For the remote procedure call (RPC) application protocol, configure an RPC program number.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc]
user@host# set rpc-program-number rpc-program-number
```

7. (Optional) Configure the SNMP command for packet matching.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc]
user@host# set snmp-command snmp-command
```

8. (Optional) For the traceroute application protocol, configure the traceroute time-to-live (TTL) threshold value. This value sets the acceptable level of network penetration for trace routing.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc]
user@host# set ttl-threshold ttl-threshold
```

9. (Optional) Enter configuration mode for the protocol attribute.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc]
user@host# edit proto-attr
```

10. (Optional) For the ICMP protocol, configure the ICMP packet type.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc proto-attr]
user@host# set icmp-type icmp-type
```

11. (Optional) For the ICMP protocol, configure the ICMP code.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc proto-attr]
user@host# set icmp-code icmp-code
```

12. (Optional) Enter the destination port configuration.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc proto-attr]
user@host# edit destination-port port
```

13. (Optional) Configure the TCP or UDP destination port.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc proto-attr destination-port port]
user@host# set from-port from-port
```

14. (Optional) Enter the source port configuration.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc proto-attr destination-port port]
user@host# up
```

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc proto-attr]
user@host# edit source-port port
```

15. (Optional) Configure the TCP or UDP source port.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc proto-attr source-port port]
user@host# set from-port from-port
```

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc proto-attr source-port port]
user@host# up
```

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc proto-attr]
user@host# up
```

16. (Optional) Verify the application protocol condition configuration.

```
[edit policies group junos list staticnat rule nat traffic-condition ctc
 application-protocol-condition apc]
user@host# show
protocol ip;
application-protocol dce_rpc;
idle-timeout 900;
dce-rpc-uuid dce_rpc;
snmp-command get;
ttl-threshold 25;
proto-attr {
  icmp-type icmpType;
  icmp-code icmpCode;
  destination-port {
    port {
      from-port 11..655;
    }
  }
}
```

```

    }
    source-port {
        port {
            from-port service_port;
        }
    }
}

```

- Related Topics**
- Using Map Expressions in Application Protocol Conditions on page 147
 - Before You Configure Classify-Traffic Conditions on page 115
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring Application Protocol Conditions (C-Web Interface)

Using Map Expressions in Application Protocol Conditions

The application protocol condition is a case in which you might use a map expression to define multiple attributes in one option—the **application-protocol** option. Maps are a list of attributeName = value pairs separated by commas and enclosed in curly brackets. For example, the map {applicationProtocol = “ftp”, sourcePort = 123, inactivityTimeout = 60} supplies the application protocol, source port, and inactivity timeout in one option. “

Another map {applicationType = “tcp”, inactivityTimeout = 60, destinationPort = 80} supplies the protocol, inactivity timeout, and destination port.

You can also create a local parameter, add a map expression as the default value of the parameter, and then enter the local parameter in the **application-protocol** option.

- Related Topics**
- Overview of Global and Local Parameters on page 81
 - Before You Configure SRC Policies on page 105
 - Configuring Application Protocol Conditions (SRC CLI) on page 144
 - Configuring Protocol Conditions (SRC CLI) on page 123
 - Configuring Protocol Conditions with Ports (SRC CLI) on page 124

Configuring QoS Conditions (SRC CLI)

You can create QoS conditions within JUNOS scheduler policy rules. Use the following configuration statements to configure a QoS condition:

```

policies group name list name rule name qos-condition name {
    forwarding-class forwarding-class ;
    description description ;
}

```

To create a QoS condition:

1. From configuration mode, enter the QoS condition configuration. For example:

```
user@host# edit policies group junos list qos rule pr qos-condition qc
```

2. (Optional) Configure the forwarding class to match.

```
[edit policies group junos list qos rule pr qos-condition qc]
user@host# set forwarding-class forwarding-class
```

3. (Optional) Enter a description of the QoS condition.

```
[edit policies group junos list qos rule pr qos-condition qc]
user@host# set description description
```

4. (Optional) Verify the QoS condition configuration.

```
[edit policies group junos list qos rule pr qos-condition qc]
user@host# show
forwarding-class assured-forwarding;
description "QoS condition for QoS scheduling";
```

- Related Topics**
- Delivering QoS Services in a Cable Environment on page 73
 - Policy Information Model on page 62
 - Configuring QoS Profile Attachment Actions (SRC CLI) on page 172
 - Configuring QoS Conditions with the C-Web Interface

Configuring Actions

Actions define the action taken on packets that match conditions in a policy rule. You create actions within policy rules.

Topics that discuss how to configure the action include:

- Configuring Color Actions (SRC CLI) on page 149
- Configuring Color Mark Actions (SRC CLI) on page 150
- Configuring DOCSIS Actions (SRC CLI) on page 151
- Configuring Exception Application Actions (SRC CLI) on page 156
- Configuring Filter Actions (SRC CLI) on page 156
- Configuring FlowSpec Actions (SRC CLI) on page 157
- Configuring Forward Actions (SRC CLI) on page 159
- Configuring Forwarding Class Actions (SRC CLI) on page 160
- Configuring GateSpec Actions (SRC CLI) on page 161
- Configuring HTTP Redirect Actions (SRC CLI) on page 162
- Configuring Loss Priority Actions (SRC CLI) on page 163
- Configuring Mark Actions (SRC CLI) on page 164
- Configuring NAT Actions (SRC CLI) on page 165

- Configuring Next-Hop Actions (SRC CLI) on page 166
- Configuring Next-Interface Actions (SRC CLI) on page 167
- Configuring Next-Rule Actions (SRC CLI) on page 168
- Configuring Policer Actions (SRC CLI) on page 169
- Configuring the Packet Action for the Policer Action (SRC CLI) on page 170
- Configuring QoS Profile Attachment Actions (SRC CLI) on page 172
- Configuring Rate-Limit Actions (SRC CLI) on page 173
- Configuring Hierarchical Rate-Limit Actions (SRC CLI) on page 175
- Configuring Types of Actions for Rate-Limit Actions (SRC CLI) on page 178
- Configuring Reject Actions (SRC CLI) on page 180
- Configuring Routing Instance Actions (SRC CLI) on page 181
- Configuring Scheduler Actions (SRC CLI) on page 182
- Configuring Drop Profiles (SRC CLI) on page 184
- Configuring Service Class Name Actions (SRC CLI) on page 185
- Configuring Stateful Firewall Actions (SRC CLI) on page 186
- Configuring Template Activation Actions (SRC CLI) on page 188
- Configuring Traffic-Class Actions (SRC CLI) on page 189
- Configuring Traffic-Mirror Actions (SRC CLI) on page 190
- Configuring Traffic-Shape Actions (SRC CLI) on page 191
- Configuring User Packet Class Actions (SRC CLI) on page 192

Configuring Color Actions (SRC CLI)

You can configure color actions for JUNOS policy rules. The type of action that you can create depends on the type of policy rule.

Use the following configuration statements to configure color actions:

```
policies group name list name rule name color name {
    color color;
    description description;
}
```

To configure a color action:

1. From configuration mode, enter the color action configuration. For example, in this procedure, *ca* is the name of the color action.

```
user@host# edit policies group junose_filter list in rule pr color ca
```

2. (Optional) Configure the color that is applied to a packet when it passes through the router.

```
[edit policies group junose_filter list in rule pr filter ca]
user@host# set color color
```

3. (Optional) Enter a description for the color action.

```
[edit policies group junose_filter list in rule pr filter ca]
user@host# set description description
```

4. (Optional) Verify the color action configuration.

```
[edit policies group junose_filter list in rule pr filter ca]
user@host# show
color green;
description "Color action for JUNOSe IPv6 policies";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Delivering QoS Services in a Cable Environment on page 73
 - Overview of Global and Local Parameters on page 81
 - Parameter Types on page 82

Configuring Color Mark Actions (SRC CLI)

You can configure color-mark actions to apply a TOS mark value based on the packet's color. A color-mark can be applied to a packet after it exits a rate-limit hierarchy.

The type of action that you can create depends on the type policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure color actions:

```
policies group name list name rule name color-mark name {
  green-mark green-mark;
  yellow-mark yellow-mark;
  red-mark red-mark;
  mask mask;
  description description;
}
```

To configure a color mark action:

1. From configuration mode, enter the color mark action configuration. For example, in this procedure, *cma* is the name of the color action.

```
[edit]
user@host# edit policies group junose_filter list in rule pr color cma
```

2. (Optional) Specify the green-mark value.

```
[edit policies group junose_filter list in rule pr filter cma]
user@host# set green-mark green-mark
```

3. (Optional) Specify the yellow-mark value.

```
[edit policies group junose_filter list in rule pr filter cma]
```



```
user@host# set yellow-mark yellow-mark
```

4. (Optional) Specify the red-mark value.

```
[edit policies group junose_filter list in rule pr filter cma]
user@host# set red-mark red-mark
```

5. Specify the mask value to use when marking packets.

```
[edit policies group junose_filter list in rule pr filter cma]
user@host# set mask mask
```

6. (Optional) Enter a description for the color mark action.

```
[edit policies group junose_filter list in rule pr filter cma]
user@host# set description description
```

7. (Optional) Verify the color action mark configuration.

```
[edit policies group junose_filter list in rule pr filter cma]
user@host# show
description mark green packets;
green-mark 100;
mask 255;
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Hierarchical Rate-Limit Actions (SRC CLI) on page 175
 - Configuring Rate-Limit Actions (SRC CLI) on page 173
 - Configuring Color Actions (SRC CLI) on page 149
 - Dynamic Bandwidth Allocation Among Traffic Flows in SRC Policy Lists for JUNOS Routing Platforms on page 71

Configuring DOCSIS Actions (SRC CLI)

You can configure Data over Cable Service Interface Specifications (DOCSIS) actions for *PacketCable Multimedia Specification* (PCMM) policy rules. The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure DOCSIS actions. Use the configuration statement for the service flow scheduling type that you want to use for the DOCSIS action. The types are best effort, downstream, non-real-time polling service, real-time polling service, unsolicited grant service, unsolicited grant service with activity detection, or parameter.

```
policies group name list name rule name docsis-best-effort name {
  traffic-priority traffic-priority ;
  request-transmission-policy request-transmission-policy ;
  maximum-sustained-rate maximum-sustained-rate ;
```

```

maximum-traffic-burst maximum-traffic-burst ;
minimum-reserved-rate minimum-reserved-rate ;
assumed-minimum-res-packet-size assumed-minimum-res-packet-size ;
description description ;
}
policies group name list name rule name docsis-down-stream name {
  traffic-priority traffic-priority ;
  maximum-latency maximum-latency ;
  maximum-sustained-rate maximum-sustained-rate ;
  maximum-traffic-burst maximum-traffic-burst ;
  minimum-reserved-rate minimum-reserved-rate ;
  assumed-minimum-res-packet-size assumed-minimum-res-packet-size ;
  description description ;
}
policies group name list name rule name docsis-non-real-time name {
  traffic-priority traffic-priority ;
  request-transmission-policy request-transmission-policy ;
  maximum-sustained-rate maximum-sustained-rate ;
  maximum-traffic-burst maximum-traffic-burst ;
  minimum-reserved-rate minimum-reserved-rate ;
  assumed-minimum-res-packet-size assumed-minimum-res-packet-size ;
  nominal-polling-interval nominal-polling-interval ;
  description description ;
}
policies group name list name rule name docsis-real-time name {
  request-transmission-policy request-transmission-policy ;
  maximum-sustained-rate maximum-sustained-rate ;
  maximum-traffic-burst maximum-traffic-burst ;
  minimum-reserved-rate minimum-reserved-rate ;
  assumed-minimum-res-packet-size assumed-minimum-res-packet-size ;
  nominal-polling-interval nominal-polling-interval ;
  tolerated-poll-jitter tolerated-poll-jitter ;
  description description ;
}
policies group name list name rule name docsis-unsolicited-grant name {
  request-transmission-policy request-transmission-policy ;
  grant-size grant-size ;
  grants-per-interval grants-per-interval ;
  tolerated-grant-jitter tolerated-grant-jitter ;
  nominal-grant-interval nominal-grant-interval ;
  description description ;
}
policies group name list name rule name docsis-unsolicited-grant-ad name {
  request-transmission-policy request-transmission-policy ;
  nominal-polling-interval nominal-polling-interval ;
  grant-size grant-size ;
  grants-per-interval grants-per-interval ;
  tolerated-grant-jitter tolerated-grant-jitter ;
  nominal-grant-interval nominal-grant-interval ;
  description description ;
}
policies group name list name rule name docsis-param name {
  service-flow-type service-flow-type ;
  traffic-priority traffic-priority ;
  request-transmission-policy request-transmission-policy ;

```

```

maximum-sustained-rate maximum-sustained-rate ;
maximum-traffic-burst maximum-traffic-burst ;
minimum-reserved-rate minimum-reserved-rate ;
assumed-minimum-res-packet-size assumed-minimum-res-packet-size ;
maximum-latency maximum-latency ;
nominal-polling-interval nominal-polling-interval ;
tolerated-poll-jitter tolerated-poll-jitter ;
grant-size grant-size ;
grants-per-interval grants-per-interval ;
tolerated-grant-jitter tolerated-grant-jitter ;
nominal-grant-interval nominal-grant-interval ;
description description ;
}

```

To configure a DOCSIS action:

1. From configuration mode, enter the DOCSIS action configuration. For example, in this procedure, DOCSISParameter is the name of the DOCSIS action.

```

user@host# edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter

```

2. Assign a parameter as the service flow scheduling type.

Before you assign a parameter, you must create a parameter of type trafficProfileType and commit the parameter configuration.

```

[edit policies group pcmm list DocsisParameter rule in docsis-param
 DOCSISParameter]
user@host# set service-flow-type service-flow-type

```

3. (Optional) Configure a priority for the service flow. If two traffic flows are identical in all QoS parameters except priority, the higher-priority service flow is given preference.

```

[edit policies group pcmm list DocsisParameter rule in docsis-param
 DOCSISParameter]
user@host# set traffic-priority traffic-priority

```

4. (Optional) Configure the request transmission policy, which is the interval usage code that the cable modem uses for upstream transmission requests and packet transmissions for this service flow. It also specifies whether requests can be piggybacked with data.

- For data packets transmitted on this service flow, this option also specifies whether packets can be concatenated, fragmented, or have their payload headers suppressed.
- For UGS service flows, this option also specifies how to treat packets that do not fit into the UGS grant.

```

[edit policies group pcmm list DocsisParameter rule in docsis-param
 DOCSISParameter]
user@host# set request-transmission-policy request-transmission-policy

```

5. (Optional) Configure the maximum sustained rate at which traffic can operate over the service flow.

```
[edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter]
user@host# set maximum-sustained-rate maximum-sustained-rate
```

6. (Optional) Configure the maximum burst size for the service flow. This option has no effect unless you configure a nonzero value for the maximum sustained rate.

```
[edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter]
user@host# set maximum-traffic-burst maximum-traffic-burst
```

7. (Optional) Configure the guaranteed minimum rate that is reserved for the service flow.

```
[edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter]
user@host# set minimum-reserved-rate minimum-reserved-rate
```

8. (Optional) Configure the assumed minimum packet size for which the minimum reserved traffic rate is provided. If a packet is smaller than the assumed minimum packet size, the software treats the packet as if its size is equal to the value specified in this option.

```
[edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter]
user@host# set assumed-minimum-res-packet-size
assumed-minimum-res-packet-size
```

9. (Optional) Configure the maximum latency for downstream service flows. It is the maximum latency for a packet that passes through the CMTS device, from the time that the CMTS device's network side interface receives the packet until the CMTS device forwards the packet on its radio frequency (RF) interface.

```
[edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter]
user@host# set maximum-latency maximum-latency
```

10. (Optional) Configure the nominal interval between successive unicast request opportunities for this service flow.

```
[edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter]
user@host# set nominal-polling-interval nominal-polling-interval
```

11. (Optional) Configure the maximum amount of time that unicast request intervals can be delayed beyond the nominal polling interval.

```
[edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter]
```

```
user@host# set tolerated-poll-jitter tolerated-poll-jitter
```

12. (Optional) Configure the size of the individual data grants provided to the service flow.

```
[edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter]
user@host# set grant-size grant-size
```

13. (Optional) Configure the actual number of data grants given to the service flow during each nominal grant interval.

```
[edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter]
user@host# set grants-per-interval grants-per-interval
```

14. (Optional) Configure the maximum amount of time that the transmission opportunities can be delayed beyond the nominal grant interval.

```
[edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter]
user@host# set tolerated-grant-jitter tolerated-grant-jitter
```

15. (Optional) Configure the nominal interval between successive unsolicited data grant opportunities for this service flow.

```
[edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter]
user@host# set nominal-grant-interval nominal-grant-interval
```

16. (Optional) Enter a description for the filter action.

```
[edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter]
user@host# set description description
```

17. (Optional) Verify the DOCSIS action configuration.

```
[edit policies group pcmm list DocsisParameter rule in docsis-param
DOCSISParameter]
user@host# show
service-flow-type action;
traffic-priority 1;
request-transmission-policy 1;
maximum-sustained-rate 1500;
maximum-traffic-burst 3044;
minimum-reserved-rate 1240;
assumed-minimum-res-packet-size 124;
description "DOCSIS parameter action with a parameter service flow
scheduling type";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Overview of Global and Local Parameters on page 81

- Delivering QoS Services in a Cable Environment on page 73
- Configuring DOCSIS Actions (C-Web Interface)
- Parameter Types on page 82

Configuring Exception Application Actions (SRC CLI)

Use the following statements to configure policy actions that specify exceptions to a rule to identify the client application that is a destination for packets.

```
policies group name list name rule name exception-application name {
    application-type application-type ;
    description description ;
}
```

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

To configure an exception action in a policy rule to specify that traffic be sent to a client application:

1. From configuration mode, enter the exception application action configuration. For example, to name the exception application action my-http-application:

```
[edit]
user@host# edit policies group http-policy list http-list rule redirect
exception-application my-http-application
```

2. Specify the application type, such as HTTP for Web traffic.

```
[edit policies group http-policy list http-list rule redirect exception-application
my-http-application]
user@host# set application-type http
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring HTTP Redirect Actions (SRC CLI) on page 162
 - Configuring the Redirect Server (SRC CLI)
 - Before You Configure the Redirect Server on a C-Series Controller
 - Parameter Types on page 82

Configuring Filter Actions (SRC CLI)

Use this action to discard packets. You can configure filter actions for JUNOS filters and JUNOS policy rules. The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statement to configure a filter action:

```
policies group name list name rule name filter name {
```

```
description description ;
}
```

To configure a filter action:

1. From configuration mode, enter the filter action configuration. For example, in this procedure, *fa* is the name of the filter action.

```
user@host# edit policies group junos_filter list in rule pr filter fa
```

2. (Optional) Enter a description for the filter action.

```
[edit policies group junos_filter list in rule pr filter fa]
user@host# set description description
```

3. (Optional) Verify the filter action configuration.

```
[edit policies group junos_filter list in rule pr filter fa]
user@host# show
description "Filter action for JUNOS policies";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Filter Actions (C-Web Interface)
 - Configuring JUNOS Filter Conditions (SRC CLI) on page 139
 - Parameter Types on page 82

Configuring FlowSpec Actions (SRC CLI)

A FlowSpec is made up of two parts, a traffic specification (TSpec) and a service request specification (RSpec). The TSpec describes the traffic requirements for the flow, and the RSpec specifies resource requirements for the desired service. You can configure FlowSpec actions for PCMM policy rules.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure FlowSpec actions:

```
policies group name list name rule name flow-spec name {
  service-type service-type ;
  token-bucket-rate token-bucket-rate ;
  token-bucket-size token-bucket-size ;
  peak-data-rate peak-data-rate ;
  minimum-policed-unit minimum-policed-unit ;
  maximum-packet-size maximum-packet-size ;
  rate rate ;
  slack-term slack-term ;
  description description ;
}
```

To configure a FlowSpec action:

1. From configuration mode, enter the FlowSpec action configuration. For example in this procedure, *fsa* is the name of the FlowSpec action.

```
user@host# edit policies group pcmm list TrafficProfileFlowSpec rule pr  
flow-spec fsa
```

2. (Optional) Configure the type of FlowSpec service as either *controlled_load_service* or *guaranteed_service*. The FlowSpec options available for configuration change depending on the type of service that you select:

- Controlled load services can contain only TSpec parameters.
- Guaranteed services can contain both TSpec and RSpec parameters.

```
[edit policies group pcmm list TrafficProfileFlowSpec rule pr flow-spec fsa]  
user@host# set service-type service-type
```

3. (Optional TSpec parameter) Configure the guaranteed minimum rate that is reserved for the service flow.

```
[edit policies group pcmm list TrafficProfileFlowSpec rule pr flow-spec fsa]  
user@host# set token-bucket-rate token-bucket-rate
```

4. (Optional TSpec parameter) Configure the maximum burst size for the service flow.

```
[edit policies group pcmm list TrafficProfileFlowSpec rule pr flow-spec fsa]  
user@host# set token-bucket-size token-bucket-size
```

5. (Optional TSpec parameter) Configure the amount of bandwidth over the committed rate that is allocated to accommodate excess traffic flow over the committed rate.

```
[edit policies group pcmm list TrafficProfileFlowSpec rule pr flow-spec fsa]  
user@host# set peak-data-rate peak-data-rate
```

6. (Optional TSpec parameter) Configure the assumed minimum-reserved-rate packet size. If a packet is smaller than the minimum policed unit, the software treats the packet as if its size is equal to the value specified in this option.

```
[edit policies group pcmm list TrafficProfileFlowSpec rule pr flow-spec fsa]  
user@host# set minimum-policed-unit minimum-policed-unit
```

7. (Optional TSpec parameter) Configure the maximum packet size for the FlowSpec.

```
[edit policies group pcmm list TrafficProfileFlowSpec rule pr flow-spec fsa]  
user@host# set maximum-packet-size maximum-packet-size
```

8. (Optional RSpec parameter) Configure the average rate.

```
[edit policies group pcmm list TrafficProfileFlowSpec rule pr flow-spec fsa]  
user@host# set rate rate
```


9. (Optional RSpec parameter) Configure the amount of slack in the bandwidth reservation that can be used without redefining the reservation.

```
[edit policies group pcmm list TrafficProfileFlowSpec rule pr flow-spec fsa]
user@host# set slack-term slack-term
```

10. (Optional) Configure a description for the FlowSpec action.

```
[edit policies group pcmm list TrafficProfileFlowSpec rule pr flow-spec fsa]
user@host# set description description
```

11. (Optional) Verify the FlowSpec action configuration.

```
[edit policies group pcmm list TrafficProfileFlowSpec rule pr flow-spec
fsa]
user@host# show
service-number guaranteed_service;
token-bucket-rate bucketRate;
token-bucket-size bucketDepth;
peak-data-rate peakRate;
minimum-policed-unit minPolicedUnit;
rate reservedRate;
slack-term slackTerm;
description "FlowSpec guaranteed service";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Delivering QoS Services in a Cable Environment on page 73
 - Configuring FlowSpec Actions (C-Web Interface)
 - Parameter Types on page 82

Configuring Forward Actions (SRC CLI)

Use this action to forward packets, such as packets that are sent by means of a routing table. You can configure forward actions for JUNOS filters and JUNOS policy rules.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statement to configure forward actions:

```
policies group name list name rule name forward name {
  description description ;
}
```

To configure a forward action:

1. From configuration mode, enter the forward action configuration. For example, in this procedure, fwdAction is the name of the forward action.

```
user@host# edit policies group junose list forward rule pr forward fwdAction
```

2. (Optional) Enter a description for the forward action.

```
[edit policies group junose list forward rule pr forward fwdAction]
user@host# set description description
```

3. (Optional) Verify the forward action configuration.

```
[edit policies group junose list forward rule pr forward fwdAction]
user@host# show
description "JUNOS Forward Action";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Forward Actions (C-Web Interface)
 - Parameter Types on page 82

Configuring Forwarding Class Actions (SRC CLI)

You can configure forwarding class actions for JUNOS filter policy rules. The forwarding class action causes the router to assign a forwarding class to packets that match the associated classify-traffic condition.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure a forwarding class action:

```
policies group name list name rule name forwarding-class name {
  forwarding-class forwarding-class ;
  description description ;
}
```

To configure a forwarding class action:

1. From configuration mode, enter the forwarding class action configuration. For example, in this procedure, fca is the name of the forwarding class action.

```
user@host# edit policies group bod list input rule pr forwarding-class fca
```

2. (Optional) Configure the name of the forwarding class assigned to packets.

```
[edit policies group bod list input rule pr forwarding-class fca]
user@host# set forwarding-class forwarding-class
```

3. (Optional) Enter a description for the forwarding class action.

```
[edit policies group bod list input rule pr forwarding-class fca]
user@host# set description description
```

4. (Optional) Verify the forwarding class action configuration.

```
[edit policies group bod list input rule pr forwarding-class fca]
user@host# show
forwarding-class fc_expedited;
description "Expedited forwarding class";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Forward Actions (SRC CLI) on page 159
 - Configuring Forwarding Class Actions (C-Web Interface)
 - Parameter Types on page 82

Configuring GateSpec Actions (SRC CLI)

You can configure GateSpec actions for PCMM policy rules. See “Policy Information Model” on page 62 for more information.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure GateSpec actions:

```
policies group name list name rule name gate-spec name {
  session-class-id-priority session-class-id-priority ;
  session-class-id-preemption session-class-id-preemption ;
  session-class-id-configurable session-class-id-configurable ;
  description description ;
}
```

To configure a GateSpec action:

1. From configuration mode, enter the GateSpec action configuration. For example, in this procedure, *gsa* is the name of the GateSpec action.

```
user@host# edit policies group pcmm list GateSpec rule pr gate-spec gsa
```

2. (Optional) Configure the priority bits in the session class ID. The priority describes the relative importance of the session as compared with other sessions generated by the same policy decision point.

```
[edit policies group pcmm list GateSpec rule pr gate-spec gsa]
user@host# set session-class-id-priority session-class-id-priority
```

3. (Optional) Configure the preemption bit in the session class ID. Use the preemption bit to allocate bandwidth to lower-priority sessions.

```
[edit policies group pcmm list GateSpec rule pr gate-spec gsa]
user@host# set session-class-id-preemption session-class-id-preemption
```

4. (Optional) Configure the configurable bit in the session class ID.

```
[edit policies group pcmm list GateSpec rule pr gate-spec gsa]
```

```
user@host# set session-class-id-configurable session-class-id-configurable
```

5. (Optional) Enter a description for the GateSpec action.

```
[edit policies group pcmm list GateSpec rule pr gate-spec gsa]
user@host# set description description
```

6. (Optional) Verify the GateSpec action configuration.

```
[edit policies group pcmm list GateSpec rule pr gate-spec gsa]
user@host# show
session-class-id-priority 5;
session-class-id-preemption 0;
session-class-id-configurable 5
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Parameter Types on page 82

Configuring HTTP Redirect Actions (SRC CLI)

Use the following statements to configure policy actions to redirect Web traffic to a specified URL.

```
policies group name list name rule name http-redirect name {
  subscriber-url "\"http:// URL \"";
  description description ;
}
```

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

To configure an HTTP redirect action in a policy rule to specify that Web traffic be sent to a specified URL:

1. From configuration mode, enter the HTTP redirect action configuration. For example, to name the HTTP redirect action residential-portal:

```
[edit]
user@host# edit policies group http list http-list rule redirect http-redirect
residential-portal
```

2. Specify the destination URL to which traffic will be redirected. For example, to redirect the traffic to www.new.com:

```
[edit policies group http list http-list rule redirect http-redirect residential-portal]
user@host# subscriber-url "\"http://www.new.com\""
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Exception Application Actions (SRC CLI) on page 156
 - Before You Configure the Redirect Server on a C-Series Controller

- Configuring a Redundant Redirect Server (SRC CLI)
- Parameter Types on page 82

Configuring Loss Priority Actions (SRC CLI)

You can configure loss priority actions for JUNOS filter policy rules. The loss priority action causes the router to assign a packet loss priority to packets that match the associated classify-traffic condition.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure loss priority actions:

```
policies group name list name rule name loss-priority name {
  loss-priority loss-priority ;
  description description ;
}
```

To configure a loss priority action:

1. From configuration mode, enter the loss priority action configuration. For example, in this procedure, lpa is the name of the loss priority action.

```
user@host# edit policies group junos list lossPriority rule pr loss-priority lpa
```

2. (Optional) Configure the packet loss priority.

```
[edit policies group junos list lossPriority rule pr loss-priority lpa]
user@host# set loss-priority loss-priority
```

3. (Optional) Enter a description for the loss priority action.

```
[edit policies group junos list lossPriority rule pr loss-priority lpa]
user@host# set description description
```

4. (Optional) Verify the loss priority action configuration.

```
[edit policies group junos list lossPriority rule pr loss-priority lpa]
user@host# show
loss-priority high_priority;
description "Loss Priority set to high";
```

Related Topics

- Before You Configure SRC Policies on page 105
- Delivering QoS Services in a Cable Environment on page 73
- Configuring Loss Priority Actions (C-Web Interface)

Configuring Mark Actions (SRC CLI)

Use this action to mark packets. You can configure mark actions for JUNOS and PCMM policy rules.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.



NOTE: If you configure a more than one mark action in a rate limit, for example, for a committed, conformed, or exceed action, configure the same mask for each action. If you use different masks, the results can be unpredictable.

If you configure a more than one mark action in a rate limit, for example, for a committed, conformed, or exceed action, configure the same mask for each action. If you use different masks, the results can be unpredictable.

Use the following configuration statements to configure a mark action:

```

policies group name list name rule name mark name {
    description description ;
}
policies group name list name rule name mark name info {
    value value ;
    mask mask ;
}

```

To configure a mark action:

1. From configuration mode, enter the mark action configuration. For example, in this procedure, markAction is the name of the mark action.

```
user@host# edit policies group junose list mark rule pr mark markAction
```

2. (Optional) Enter a description for the mark action.

```

[edit policies group junose list mark rule pr mark markAction]
user@host# set description description

```

3. (Optional) Configure the mark value.

```

[edit policies group junose list mark rule pr mark markAction]
user@host# set info value value

```

4. (Optional) Configure the mark mask.

```

[edit policies group junose list mark rule pr mark markAction]
user@host# set info mask mask

```

Supported values for the mask are 224 for precedence bits, 252 for DS bits, and 255 to mask all bits.

5. (Optional) Verify the mark action configuration.

```
[edit policies group junose list mark rule pr mark markAction]
user@host# show
info {
  mark-value 10;
  mask 255;
}
description "Mark action";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Mark Actions (C-Web Interface)

Configuring NAT Actions (SRC CLI)

You can configure NAT actions for JUNOS ASP policy rules. The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure NAT actions:

```
policies group name list name rule name nat name {
  translation-type translation-type ;
  description description ;
}
policies group name list name rule name nat name port {
  from-port from-port ;
}
policies group name list name rule name nat name ip-network group-network {
  network-specifier network-specifier ;
}
```

To configure a NAT action:

1. From configuration mode, enter the NAT action configuration. For example, in this procedure, *natAction* is the name of the NAT action.

```
user@host# edit policies group junos list nat rule pr nat natAction
```

2. (Optional) Configure the type of network address translation that is used.

```
[edit policies group junos list nat rule pr nat natAction]
user@host# set translation-type translation-type
```

3. (Optional) Enter a description for the NAT action.

```
[edit policies group junos list nat rule pr nat natAction]
user@host# set description description
```

4. (Optional) Configure the port range to restrict port translation when the NAT translation type is configured in dynamic-source mode.

```
[edit policies group junos list nat rule pr nat natAction]
user@host# set port from-port from-port
```

5. (Optional) Configure the IP address ranges.

```
[edit policies group junos list nat rule pr nat natAction]
user@host# set ip-network group-network network-specifier network-specifier
```

6. (Optional) Verify the NAT action configuration.

```
[edit policies group junos list nat rule pr nat natAction]
user@host# show
translation-type "source dynamic";
ip-network {
  group-network {
    network-specifier 192.168.1.100/32;
  }
}
port {
  from-port 2010..2020;
}
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring NAT Actions (C-Web Interface)
 - Configuring Mark Actions (SRC CLI) on page 164
 - Parameter Types on page 82

Configuring Next-Hop Actions (SRC CLI)

Use this action for the ingress side of the interface to specify the next IP address where the classified packets should go. You can configure next-hop actions for JUNOS filters and JUNOS policy rules.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure the next-hop action.

```
policies group name list name rule name next-hop name {
  next-hop-address next-hop-address ;
  description description ;
}
```

To configure a next-hop action:

1. From configuration mode, enter the next-hop action configuration. For example, in this procedure, *nha* is the name of the next-hop action.

```
user@host# edit policies group junose list nexthop-to-ssp rule to-ssp next-hop  
nha
```

2. (Optional) Configure the next IP address where the classified packets should go.

```
[edit policies group junose list nexthop-to-ssp rule to-ssp next-hop nha]
user@host# set next-hop-address next-hop-address
```


3. (Optional) Enter a description for the next-hop action.

```
[edit policies group junose list nexthop-to-ssp rule to-ssp next-hop nha]
user@host# set description description
```

4. (Optional) Verify the next-hop action configuration.

```
[edit policies group junose list nexthop-to-ssp rule to-ssp next-hop nha]
user@host# show
next-hop-address virtual_ipAddress;
description "Next hop action";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Next-Hop Actions (C-Web Interface)
 - Configuring Next-Interface Actions (SRC CLI) on page 167
 - Configuring Next-Rule Actions (SRC CLI) on page 168
 - Parameter Types on page 82

Configuring Next-Interface Actions (SRC CLI)

Use this action to forward packets to a particular interface and/or a next-hop address. You can configure next-interface actions for JUNOS filters and JUNOS policy rules. On JUNOS routers, you can use this action for both ingress and egress parts of the interface.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure next-interface actions:

```
policies group name list name rule name next-interface name {
  interface-specifier interface-specifier ;
  next-hop-address next-hop-address ;
  description description ;
}
```

To configure a next-interface action:

1. From configuration mode, enter the next-interface action configuration. For example, in this procedure, `nextInterface` is the name of the next-interface action.

```
user@host# edit policies group redirect list input rule redirect next-interface
nextInterface
```

2. (Optional) Configure the IP interface to be used as the next interface for packets.

```
[edit policies group redirect list input rule redirect next-interface nextInterface]
user@host# set interface-specifier interface-specifier
```

3. (Optional) Configure the next IP address where the classified packets should go. This option is available only in JUNOS policy rules.

```
[edit policies group redirect list input rule redirect next-interface nextInterface]
user@host# set next-hop-address next-hop-address
```

4. (Optional) Enter a description for the next-interface action.

```
[edit policies group redirect list input rule redirect next-interface nextInterface]
user@host# set description description
```

5. (Optional) Verify the next-interface action configuration.

```
[edit policies group redirect list input rule redirect next-interface
nextInterface]
user@host# show
interfaceSpec "name='fastethernet3/0'";
next-hop-address 10.10.227.3;
description "Next-interface action for redirect policy";
```

Related Topics

- Before You Configure SRC Policies on page 105
- Configuring Next-Interface Actions (C-Web Interface)
- Configuring Next-Hop Actions (SRC CLI) on page 166
- Configuring Next-Rule Actions (SRC CLI) on page 168
- Parameter Types on page 82

Configuring Next-Rule Actions (SRC CLI)

You can configure next-rule actions for JUNOS filter policy rules. If a packet matches the classify-traffic condition, the next-rule action causes the router to continue to the next rule in the policy list for evaluation.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statement to configure next-rule actions.

```
policies group name list name rule name next-rule name {
  description description ;
}
```

To configure a next-rule action:

1. From configuration mode, enter the next-rule action configuration. For example, in this procedure, nra is the name of the next-rule action.

```
user@host# edit policies group junos list filter rule next next-rule nra
```

2. (Optional) Enter a description for the next-rule action.

```
[edit policies group junos list filter rule next next-rule nra]
user@host# set description description
```

3. (Optional) Verify the next-rule action configuration.

```
[edit policies group junos list filter rule next next-rule nra]
user@host# show configuration policies group junos list filter rule next

next-rule nra
description "Next-rule action";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Next-Rule Actions (C-Web Interface)
 - Configuring Next-Hop Actions (SRC CLI) on page 166
 - Configuring Next-Interface Actions (SRC CLI) on page 167
 - Parameter Types on page 82

Configuring Policer Actions (SRC CLI)

The policer action specifies rate and burst size limits and the action taken if a packet exceeds those limits. You can create policer actions in JUNOS policer and JUNOS filter policy rules.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure policer actions:

```
policies group name list name rule name policer name {
  bandwidth-limit bandwidth-limit ;
  bandwidth-limit-unit bandwidth-limit-unit ;
  burst burst ;
  description description ;
}
```

To configure a policer action:

1. From configuration mode, enter the policer action configuration. For example, in this procedure, *pa* is the name of the policer action.

```
user@host# edit policies group junos list firewallFilterPolicer rule pr policer pa
```

2. (Optional) Configure the traffic rate that, if exceeded, causes the router to take the indicated packet action.

```
[edit policies group junos list firewallFilterPolicer rule pr policer pa]
user@host# set bandwidth-limit bandwidth-limit
```

3. (Optional) Configure the type of value entered for bandwidth limit.

```
[edit policies group junos list firewallFilterPolicer rule pr policer pa]
user@host# set bandwidth-limit-unit bandwidth-limit-unit
```

4. (Optional) Configure the maximum burst size. The minimum recommended value is the maximum transmission unit (MTU) of the IP packets being policed.

```
[edit policies group junos list firewallFilterPolicer rule pr policer pa]
user@host# set burst burst
```

5. (Optional) Enter a description for the policer action.

```
[edit policies group junos list firewallFilterPolicer rule pr policer pa]
user@host# set description description
```

6. (Optional) Verify the policer action configuration.

```
[edit policies group junos list firewallFilterPolicer rule pr policer pa]
user@host# show
bandwidth-limit 1048576;
bandwidth-limit-unit bps;
burst 15000;
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring the Packet Action for the Policer Action (SRC CLI) on page 170
 - Configuring Policer Actions (C-Web Interface)
 - Parameter Types on page 82

Configuring the Packet Action for the Policer Action (SRC CLI)

The packet action specifies the action taken on a packet that exceeds its rate limits. You configure packet actions within policer actions.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure a packet action:

```
policies group name list name rule name policer name packet-action name ...
policies group name list name rule name policer name packet-action name
  forwarding-class {
    forwarding-class forwarding-class ;
  }
policies group name list name rule name policer name packet-action name
  loss-priority {
    loss-priority loss-priority ;
  }
policies group name list name rule name policer name packet-action name
  parameter {
    action action ;
  }
```

To configure a packet action:

1. From configuration mode, enter the packet action configuration. For example, in this procedure, `pktAction` is the name of the packet action.

```
user@host# edit policies group junos list firewallFilterPolicer rule pr policer pa
packet-action pktAction
```

2. (Optional) Configure the action to take on packets that exceed the bandwidth limit configured in the policer action.

- Filter—Packets are discarded.

```
[edit policies group junos list firewallFilterPolicer rule pr policer pa
packet-action pktAction]
user@host# set filter
```

- Forwarding class—Packets are assigned to the forwarding class that you specify.

```
[edit policies group junos list firewallFilterPolicer rule pr policer pa
packet-action pktAction]
user@host# set forwarding-class forwarding-class
```

- Loss priority—Packets are assigned the loss priority that you specify.

```
[edit policies group junos list firewallFilterPolicer rule pr policer pa
packet-action pktAction]
user@host# set loss-priority loss-priority
```

- Parameter—The action specified by the parameter is applied. Before you assign a parameter, you must create a parameter of type `packetOperation` and commit the parameter configuration.

```
[edit policies group junos list firewallFilterPolicer rule pr policer pa
packet-action pktAction]
user@host# edit parameter
[edit policies group junos list firewallFilterPolicer rule pr policer pa
packet-action pktAction parameter]
user@host# set action paramAction
```

3. (Optional) Verify the policer action configuration.

```
[edit policies group junos list firewallFilterPolicer rule pr policer pa
packet-action pktAction parameter]
user@host# show
packet-action pktAction {
  parameter {
    action PolicyParameterAction;
  }
}
bandwidth-limit 1048576;
```

```
bandwidth-limit-unit bps;
burst 15000;
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Policer Actions (SRC CLI) on page 169

Configuring QoS Profile Attachment Actions (SRC CLI)

Use this action to specify the QoS profile and the QoS parameters to attach to the router interface when this action is taken. The QoS profile and the QoS parameters must be configured on the router. You can configure QoS actions for JUNOS policy rules.

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. The SRC software 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. See *Dynamically Managing QoS Profiles* .

The QoS parameters allow you to specify rates in QoS profiles as parameters instead of fixed values. The actual values for the parameters can be specified for each interface. Therefore, you can share a QoS profile among different interfaces with different rates.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure QoS profile attachment actions:

```
policies group name list name rule name qos-attach name {
  qos-profile qos-profile ;
  qos-parameters qos-parameters ;
  description description ;
}
```

To configure a QoS profile attachment action:

1. From configuration mode, enter the QoS profile attachment action configuration. For example, in this procedure, qos_vod is the name of the QoS profile attachment action.

```
user@host# edit policies group junose list qos rule input qos-attach qos_vod
```

2. Configure the name of the QoS profile to attach to the JUNOS interface when this action is taken.

```
[edit policies group junose list qos rule input qos-attach qos_vod]
user@host# set qos-profile qos-profile
```

3. (Optional) Configure the names and values of the QoS parameters to attach to the JUNOS interface when this action is taken. Use map expressions to specify multiple values.

```
[edit policies group junose list qos rule input qos-attach qos_vod]
user@host# set qos-parameters qos-parameters
```

4. (Optional) Enter a description for the QoS profile attachment action.

```
[edit policies group junose list qos rule input qos-attach qos_vod]
user@host# set description description
```

5. (Optional) Verify the QoS profile attachment action configuration.

```
[edit policies group junose list qos rule input qos-attach qos_vod]
user@host# show
qos-profile qp-vod-1024;
description "Action for QoS video-on-demand";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Delivering QoS Services in a Cable Environment on page 73
 - Configuring QoS Conditions (SRC CLI) on page 147
 - Configuring QoS Profile Attachment Actions (C-Web Interface)

Configuring Rate-Limit Actions (SRC CLI)

Use this action to define the quality of service. You can configure rate-limit actions for JUNOS policy rules.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure rate-limit actions:

```
policies group name list name rule name rate-limit name {
  type type ;
  committed-rate committed-rate ;
  committed-burst committed-burst ;
  peak-rate peak-rate ;
  peak-burst peak-burst ;
  excess-burst excess-burst ;
  color-aware;
  support-hierarchical;
  description description ;
}
```

To configure a rate-limit action:

1. From configuration mode, enter the rate-limit action configuration. For example, in this procedure, rla is the name of the rate-limit action.

```
user@host# edit policies group junose list rate-limiter rule pr rate-limit rla
```

2. (Optional) Specify that the rate-limit profile is either one rate or two rate. The rate-limit type determines the options that you can configure for a rate-limit action.

```
[edit policies group junose list rate-limiter rule pr rate-limit rla]
user@host# set type type
```

3. (Optional) Configure the target rate for the traffic that the policy covers.

```
[edit policies group junose list rate-limiter rule pr rate-limit rla]
user@host# set committed-rate committed-rate
```

4. (Optional) Configure the amount of bandwidth allocated to burst traffic in bytes.

```
[edit policies group junose list rate-limiter rule pr rate-limit rla]
user@host# set committed-burst committed-burst
```

5. (Optional) For two-rate rate-limit profiles, specify the amount of bandwidth allocated to excess traffic flow over the committed rate.

```
[edit policies group junose list rate-limiter rule pr rate-limit rla]
user@host# set peak-rate peak-rate
```

6. (Optional) For two-rate rate-limit profiles, specify the amount of bandwidth allocated to burst traffic in excess of the peak rate.

```
[edit policies group junose list rate-limiter rule pr rate-limit rla]
user@host# set peak-burst peak-burst
```

7. (Optional) For one-rate rate-limit profiles, specify the amount of bandwidth allocated to accommodate burst traffic.

```
[edit policies group junose list rate-limiter rule pr rate-limit rla]
user@host# set excess-burst excess-burst
```

8. (Optional) Set the rate limit to color aware for the rate-limit to process color values.

```
[edit policies group parentGroupDefault list twoRate parent-group PGShared]
user@host# set color-aware
```

9. (Optional) Set the rate limit to support hierarchical rate limits.

```
[edit policies group junose list rate-limiter rule pr rate-limit rla]
user@host# set support-hierarchical
```

10. (Optional) Configure the rate-limit action for traffic flows that:

- Do not exceed the committed rate
- Exceed the committed rate but remain below the peak rate.

- Exceed the peak rate.

See “Configuring Types of Actions for Rate-Limit Actions (SRC CLI)” on page 178.

11. (Optional) Enter a description for the rate-limit action.

```
[edit policies group junose list rate-limiter rule pr rate-limit rla]
user@host# set description description
```

12. (Optional) Return to the rate-limit action configuration, and verify the configuration.

```
[edit policies group junose list rate-limiter rule pr rate-limit rla]
user@host# show
committed-action {
  forward {
  }
}
conformed-action {
  forward {
  }
}
exceed-action {
  filter {
  }
}
type 1;
committed-rate 1000000;
committed-burst 125000;
excess-burst 312500;
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Rate-Limit Actions (C-Web Interface)
 - Configuring Filter Actions (SRC CLI) on page 156
 - Configuring Forward Actions (SRC CLI) on page 159

Configuring Hierarchical Rate-Limit Actions (SRC CLI)

You configure parent groups to create a hierarchy of rate limits.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure rate-limit actions:

```
policies group name list name parent-group name {
  description description;
}
policies group name list name parent-group name parent-group name {
  description description;
}
```

```

policies group name list name parent-group name rate-limit {
    type type;
    committed-rate committed-rate;
    committed-burst committed-burst;
    peak-rate peak-rate;
    peak-burst peak-burst;
    excess-burst excess-burst;
    color-aware;
}

```

To configure a parent group:

1. From configuration mode, create a parent group in a policy list; for example, PGShared.

```

[edit policies group parentGroupDefault list twoRate]
user@host# edit parent-group PGShared

```

2. Configure a rate limit for the parent group.

```

[edit policies group parentGroupDefault list twoRate parent-group PGShared]
user@host# edit rate-limit rate-limit

```

3. Specify that the rate-limit is either one rate or two rate. The rate-limit type determines the options that you can configure for a rate-limit action.

```

[edit policies group parentGroupDefault list twoRate parent-group PGShared
rate-limit]
user@host# set type type

```

4. (Optional) Configure the target rate for the traffic that the policy covers.

```

[edit policies group parentGroupDefault list twoRate parent-group PGShared
rate-limit]
user@host# set committed-rate committed-rate

```

5. (Optional) Configure the amount of bandwidth allocated to burst traffic in bytes.

```

[edit policies group parentGroupDefault list twoRate parent-group PGShared
rate-limit]
user@host# set committed-burst committed-burst

```

6. (Optional) For two-rate rate-limit profiles, specify the amount of bandwidth allocated to excess traffic flow over the committed rate.

```

[edit policies group parentGroupDefault list twoRate parent-group PGShared
rate-limit]
user@host# set peak-rate peak-rate

```

7. (Optional) For two-rate rate-limit profiles, specify the amount of bandwidth allocated to burst traffic in excess of the peak rate.

```

[edit policies group parentGroupDefault list twoRate parent-group PGShared
rate-limit]
user@host# set peak-burst peak-burst

```

8. (Optional) For one-rate rate-limit profiles, specify the amount of bandwidth allocated to accommodate burst traffic.

```
[edit policies group parentGroupDefault list twoRate parent-group PGShared]
user@host# set excess-burst excess-burst
```

9. (Optional) Set the rate limit to color aware for the rate-limit to process color values set in the TOS byte of the packet header.

```
[edit policies group parentGroupDefault list twoRate parent-group PGShared]
user@host# set color-aware
```

10. (Optional) Configure the rate-limit actions for traffic flows that:

- Do not exceed the committed rate and burst size
- Exceed the committed rate but remains below the peak burst size
- Exceed the committed rate but remains below the peak burst size

See “Configuring Types of Actions for Rate-Limit Actions (SRC CLI)” on page 178.

11. Enter a description for the parent group.

```
[edit policies group parentGroupDefault list twoRate parent-group PGShared]
user@host# set description description
```

12. (Optional) Return to the rate-limit action configuration, and verify the configuration.

```
[edit policies group parentGroupDefault list twoRate parent-group PGShared]
user@host# show
```

13. (Optional) Create another parent group within this parent group. For example:

```
[edit policies group parentgroupDefault list twoRate parent-group PGShared]
user@host# parent-group parent-group-name
user@host# set parent-group PgInternet
```

14. (Optional) Follow the instructions in this procedure for configuring the parent group.

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Rate-Limit Actions (SRC CLI) on page 173
 - Dynamic Bandwidth Allocation Among Traffic Flows in SRC Policy Lists for JUNOS Routing Platforms on page 71

Configuring Types of Actions for Rate-Limit Actions (SRC CLI)

When you configure a rate-limit action or a rate-limit hierarchy, you specify types of actions for conformed traffic, committed traffic, and exceed traffic.

Before you configure the various types of action, configure the rate limit or parent group (for rate-limit hierarchies) that is to include the actions. See

- Configuring Rate-Limit Actions (SRC CLI) on page 173
- Configuring Hierarchical Rate-Limit Actions (SRC CLI) on page 175

Use the following configuration statements to configure rate-limit actions:

```

policies group name list name rule name rate-limit name committed-action mark
  mark-info {
    value value ;
    mask mask ;
  }
policies group name list name rule name rate-limit name committed-action
  parameter {
    action action ;
  }
policies group name list name rule name rate-limit name conformed-action mark
  mark-info {
    value value ;
    mask mask ;
  }
policies group name list name rule name rate-limit name conformed-action
  parameter {
    action action ;
  }
policies group name list name rule name rate-limit name exceed-action mark
  mark-info {
    value value ;
    mask mask ;
  }
policies group name list name rule name rate-limit name exceed-action parameter
  {
    action action ;
  }

```

Use the following configuration statements to configure rate-limit actions for rate-limit hierarchies:

```

policies group name list name parent-group name rate-limit committed-action filter
policies group name list name parent-group name rate-limit committed-action forward
policies group name list name parent-group name rate-limit committed-action
  forward-conditional
policies group name list name parent-group name rate-limit committed-action
  forward-final
policies group name list name parent-group name rate-limit committed-action
  forward-unconditional
policies group name list name parent-group name rate-limit committed-action mark
  mark-info {
    value value ;
  }

```

```

    mask mask;
}
policies group name list name parent-group name rate-limit committed-action
    parameter {
        action action;
    }
policies group name list name parent-group name rate-limit conformed-action filter
policies group name list name parent-group name rate-limit conformed-action forward
policies group name list name parent-group name rate-limit conformed-action
    forward-conditional
policies group name list name parent-group name rate-limit conformed-action
    forward-final
policies group name list name parent-group name rate-limit conformed-action
    forward-unconditional
    policies group name list name parent-group name rate-limit conformed-action
        mark mark-info {
            value value;
            mask mask;
        }
    }
policies group name list name parent-group name rate-limit conformed-action
    parameter {
        action action;
    }
}
policies group name list name parent-group name rate-limit exceed-action filter
policies group name list name parent-group name rate-limit exceed-action forward
policies group name list name parent-group name rate-limit exceed-action
    forward-conditional
policies group name list name parent-group name rate-limit exceed-action
    forward-final
policies group name list name parent-group name rate-limit exceed-action mark
    mark-info {
        value value;
        mask mask;
    }
}
policies group name list name parent-group name rate-limit exceed-action
    parameter {
        action action;
    }
}

```

To specify types of actions for rate-limit actions:

- Specify an action for a rate-limit action or a rate-limit hierarchy:



NOTE: The following examples show the configuration for a committed action. You can configure the same actions for committed actions, conformed actions, and exceed actions.

- Filter (drop packet).

```

[edit policies group junose list rate-limiter rule pr rate-limit rla]
user@host# set committed-action filter

```

- Forward.

```
[edit policies group parentGroupDefault list twoRate parent-group PGShared
rate-limit]
```

```
user@host# set committed-action forward
```

- (For rate-limit hierarchies only) Forward conditionally.

```
[edit policies group parentGroupDefault list twoRate parent-group PGShared
rate-limit]
```

```
user@host# set committed-action forward-conditional
```

- (Rate-limit hierarchies only for committed and conformed actions) Forward unconditionally.

```
[edit policies group parentGroupDefault list twoRate parent-group PGShared
rate-limit]
```

```
user@host# set committed-action forward-unconditional
```

- (For rate-limit hierarchies only) Forward and exit from the rate-limit hierarchy.

```
[edit policies group parentGroupDefault list twoRate parent-group PGShared
rate-limit]
```

```
user@host# set committed-action forward-final
```

- Mark. If you select mark, enter the mark values.

```
[edit policies group junose list rate-limiter rule pr rate-limit rla]
```

```
user@host# edit committed-action mark mark-info
```

```
[edit policies group junose list rate-limiter rule pr rate-limit rla committed-action
mark mark-info]
```

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

```
[edit policies group junose list rate-limiter rule pr rate-limit rla committed-action
mark mark-info]
```

```
user@host# set mask mask
```

Make sure that masks configured with a rate limit action are the same for different mark actions.

- Parameter. Before you assign a parameter, you must create a parameter of type packetOperation and commit the parameter configuration.

```
[edit policies group junose list rate-limiter rule pr rate-limit rla
committed-action]
```

```
user@host# set committed-action parameter action action
```

- Related Topics**
- Configuring Mark Actions (SRC CLI) on page 164
 - Parameter Types on page 82

Configuring Reject Actions (SRC CLI)

You can configure reject actions for JUNOS filter policy rules. The reject action causes the router to discard a packet and send an ICMP destination unreachable message.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure reject actions:

```
policies group name list name rule name reject name {
    message-type message-type ;
    description description ;
}
```

To configure a reject action:

1. From configuration mode, enter the reject action configuration. For example, in this procedure, rejectAction is the name of the reject action.

```
user@host# edit policies group junos list filter rule rejectRule reject rejectAction
```

2. (Optional) Configure the type of ICMP destination unreachable message sent to the client.

```
[edit policies group junos list filter rule rejectRule reject rejectAction]
user@host# set message-type message-type
```

3. (Optional) Enter a description for the reject action.

```
[edit policies group junos list filter rule rejectRule reject rejectAction]
user@host# set description description
```

4. (Optional) Verify the reject action configuration.

```
[edit policies group junos list filter rule rejectRule reject rejectAction]
user@host# show
message-type network-prohibited;
description "Reject action in JUNOS filter policy";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Destination Networks (SRC CLI) on page 121
 - Parameter Types on page 82

Configuring Routing Instance Actions (SRC CLI)

You can configure routing instance actions for JUNOS filter policy rules. Use routing instance actions for filter-based forwarding to direct traffic to a specific routing instance configured on the router.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure routing instance actions:

```
policies group name list name rule name routing-inst name {
```

```

routing-instance routing-instance ;
description description ;
}

```

To configure a routing instance action:

1. From configuration mode, enter the routing instance action configuration. For example, in this procedure, *ria* is the name of the routing instance action.

```

user@host# edit policies group junos list bodVpn rule pr routing-inst ria

```

2. (Optional) Configure the routing instance to which packets are forwarded. The routing instance must be configured on the router.

```

[edit policies group junos list bodVpn rule pr routing-inst ria]
user@host# set routing-instance routing-instance

```

3. (Optional) Enter a description for the reject action.

```

[edit policies group junos list bodVpn rule pr routing-inst ria]
user@host# set description description

```

4. (Optional) Verify the routing instance action configuration.

```

[edit policies group junos list bodVpn rule pr routing-inst ria]
user@host# show
routing-instance isp2-route-table;
description "Routing Instance Action";

```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Routing Instance Actions (C-Web Interface)

Configuring Scheduler Actions (SRC CLI)

You use scheduler actions along with QoS conditions and traffic-shape actions to configure transmission scheduling and rate control. Schedulers define the priority, bandwidth, delay buffer size, rate control status, and random early detection (RED) drop profiles to be applied to a particular class of traffic. You can create scheduler actions in JUNOS scheduler policy rules.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure scheduler actions:

```

policies group name list name rule name scheduler-action name {
  buffer-size buffer-size ;
  buffer-size-unit buffer-size-unit ;
  priority priority ;
  transmit-rate transmit-rate ;
  transmit-rate-unit transmit-rate-unit ;
  exact exact ;
}

```



```

    description description ;
}

```

To configure a scheduler action:

1. From configuration mode, enter the scheduler action configuration. For example, in this procedure, *sa* is the name of the scheduler action.

```

user@host# edit policies group junos list qos rule pr scheduler-action sa

```

2. (Optional) Configure the queue transmission buffer size.

```

[edit policies group junos list qos rule pr scheduler-action sa]
user@host# set buffer-size buffer-size

```

3. (Optional) Configure the type of value that you entered for buffer size.

```

[edit policies group junos list qos rule pr scheduler-action sa]
user@host# set buffer-size-unit buffer-size-unit

```

4. (Optional) Configure the packet-scheduling priority. The priority determines the order in which an output interface transmits traffic from the queues.

```

[edit policies group junos list qos rule pr scheduler-action sa]
user@host# set priority priority

```

5. (Optional) Configure the transmit rate.

```

[edit policies group junos list qos rule pr scheduler-action sa]
user@host# set transmit-rate transmit-rate

```

6. (Optional) Configure the type of value entered for transmit rate.

```

[edit policies group junos list qos rule pr scheduler-action sa]
user@host# set transmit-rate-unit transmit-rate-unit

```

7. (Optional) Specify whether or not to enforce the exact transmission rate. Under sustained congestion, a rate-controlled queue that goes into negative credit fills up and eventually drops packets.

```

[edit policies group junos list qos rule pr scheduler-action sa]
user@host# set exact exact

```

8. (Optional) Enter a description for the scheduler action.

```

[edit policies group junos list qos rule pr scheduler-action sa]
user@host# set description description

```

9. (Optional) Verify the scheduler action configuration.

```

[edit policies group junos list qos rule pr scheduler-action sa]
user@host# show
buffer-size 85;

```

```

buffer-size-unit buffer_size_percentage;
priority low;
transmit-rate 10485760;
transmit-rate-unit rate_in_bps;
description "Scheduler action for logical interface scheduling";

```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Scheduler Actions (C-Web Interface)
 - Configuring QoS Conditions (SRC CLI) on page 147
 - Parameter Types on page 82

Configuring Drop Profiles (SRC CLI)

You configure drop profiles within scheduler actions. Drop profiles support the RED process by defining the drop probabilities across the range of delay-buffer occupancy. For a packet to be dropped, it must match the drop profile. When a packet arrives, RED checks the queue fill level. If the fill level corresponds to a nonzero drop probability, the RED algorithm determines whether to drop the arriving packet. Depending on the drop probabilities, RED might drop packets aggressively long before the buffer becomes full, or it might drop only a few packets even if the buffer is almost full.

In drop profiles you configure the queue threshold and drop probability as paired values. The values can be either percentage values (segmented) or data points (interpolated). These two alternatives enable you to configure each drop probability at up to 64 fill-level/drop-probability paired values, or to configure a profile represented as a series of line segments. For more information about configuring fill level and drop probabilities, see the JUNOS routing platform documentation.

Use the following configuration statements to configure drop profiles:

```

policies group name list name rule name scheduler-action name drop-profile
  name {
    loss-priority loss-priority ;
    protocol protocol ;
    drop-probability drop-probability ;
    drop-profile-type drop-profile-type ;
    queue-threshold queue-threshold ;
  }

```

To configure drop profiles:

1. From configuration mode, enter the drop profile configuration. For example, in this procedure, drop1 is the name of the drop profile.

```

user@host# edit policies group junos list qosWithDropProfile rule pr
scheduler-action sa drop-profile drop1

```

2. Configure the loss priority.

```
[edit policies group junos list qosWithDropProfile rule pr scheduler-action sa
drop-profile drop1]
user@host# set loss-priority loss-priority
```

3. Configure the protocol type.

```
[edit policies group junos list qosWithDropProfile rule pr scheduler-action sa
drop-profile drop1]
user@host# set protocol protocol
```

4. Configure the relationship between the fill level and drop probability.

```
[edit policies group junos list qosWithDropProfile rule pr scheduler-action sa
drop-profile drop1]
user@host# set drop-profile-type drop-profile-type
```

5. Configure the probability that a packet will be dropped.

```
[edit policies group junos list qosWithDropProfile rule pr scheduler-action sa
drop-profile drop1]
user@host# set drop-probability drop-probability
```

6. Configure the fill level of the queue.

```
[edit policies group junos list qosWithDropProfile rule pr scheduler-action sa
drop-profile drop1]
user@host# set queue-threshold queue-threshold
```

7. (Optional) Verify the drop profile configuration.

```
[edit policies group junos list qosWithDropProfile rule pr scheduler-action
sa drop-profile drop1]
user@host# show
loss-priority high_priority;
protocol any_protocol;
drop-probability "[75, 100]";
drop-profile-type interpolated;
queue-threshold "[50, 80]";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Drop Profiles (C-Web Interface)
 - Configuring Loss Priority Actions (SRC CLI) on page 163
 - Configuring Protocol Conditions (SRC CLI) on page 123

Configuring Service Class Name Actions (SRC CLI)

You can configure service class name actions for PCMM policy rules. The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure service class name actions:

```
policies group name list name rule name service-class-name name {
  service-class-name service-class-name ;
  description description ;
}
```

To configure a service class name action:

1. From configuration mode, enter the service class name action configuration. For example, in this procedure, *scna* is the name of the service class name action.

```
user@host# edit policies group pcmm list serviceClass rule pr service-class-name
scna
```

2. (Optional) Configure the name of a service class on the CMTS device that specifies QoS parameters for a service flow.

```
[edit policies group pcmm list serviceClass rule pr service-class-name scna]
user@host# set service-class-name service-class-name
```

3. (Optional) Enter a description for the service class name action.

```
[edit policies group pcmm list serviceClass rule pr service-class-name scna]
user@host# set description description
```

4. (Optional) Verify the service class name action configuration.

```
[edit policies group pcmm list serviceClass rule pr service-class-name
scna]
user@host# show configuration policies group pcmm list serviceClass rule
pr
service-class-name scna
service-class-name scn_up;
description "Service class name action for pcmm service class policy.";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Service Class Name Actions (C-Web Interface)
 - Example: Providing Premium Services on page 203

Configuring Stateful Firewall Actions (SRC CLI)

You can configure stateful firewall actions for JUNOS ASP policy rules. Stateful firewall actions specify the action to take on packets that match the classify-traffic condition.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure stateful firewall actions:

```
policies group name list name rule name stateful-firewall name {
```

```

    description description ;
}
policies group name list name rule name stateful-firewall name packet-action
  reject {
    message-type message-type ;
  }
policies group name list name rule name stateful-firewall name packet-action
  parameter {
    action action ;
  }
}

```

To configure a stateful firewall action:

1. From configuration mode, enter the stateful firewall action configuration. For example, in this procedure, *sfa* is the name of the stateful firewall action.

```

user@host# edit policies group junos list sfw rule pr stateful-firewall sfa

```

2. (Optional) Set the action to take on a packet to one of the following:

- Filter.

```

[edit policies group junos list sfw rule pr stateful-firewall sfa]
user@host# set packet-action filter

```

- Forward.

```

[edit policies group junos list sfw rule pr stateful-firewall sfa]
user@host# set packet-action forward

```

- Reject. If you set the action to reject, configure the type of ICMP destination unreachable message sent to the client.

```

[edit policies group junos list sfw rule pr stateful-firewall sfa]
user@host# set packet-action reject message-type message-type

```

- Parameter. Before you assign a parameter, you must create a parameter of type packetOperation and commit the parameter configuration.

```

[edit policies group junos list sfw rule pr stateful-firewall sfa]
user@host# set packet-action parameter action action

```

3. (Optional) Enter a description for the stateful firewall action.

```

[edit policies group junos list sfw rule pr stateful-firewall sfa]
user@host# set description description

```

4. (Optional) Verify the stateful firewall action configuration.

```

[edit policies group junos list sfw rule pr stateful-firewall sfa]
user@host# show
packet-action {
  reject {
    message-type administratively-prohibited;
  }
}

```

```

    }
    description "Stateful firewall action";

```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Stateful Firewall Actions (C-Web Interface)
 - Configuring Classify-Traffic Conditions on page 113

Configuring Template Activation Actions (SRC CLI)

Use this action to activate templates for RADIUS-enabled devices. You can configure template activation actions for AAA policy rules.

The template name and parameters are listed in the custom router template on the IMS AAA Server.

Use the following configuration statements to configure template activation actions:

```

policies group name list name rule name template-activation name {
    template-name template-name;
    description description;
}

policies group name list name rule name template-activation name variables name {
    value value;
    type type;
}

```

To configure a template activation action:

1. From configuration mode, enter the template activation action configuration. For example, in this procedure, ta is the name of the template activation action.

```

user@host# edit policies group tiered_aaa list l1 rule r1 template-activation ta

```

2. Enter the template name to activate.

```

[edit policies group tiered_aaa list l1 rule r1 template-activation ta]
user@host# set template-name template-name

```

3. (Optional) Enter a description for the template activation action.

```

[edit policies group tiered_aaa list l1 rule r1 template-activation ta]
user@host# set description description

```

4. From configuration mode, enter the parameters used by the template.

```

user@host# edit policies group tiered_aaa list l1 rule r1 template-activation ta
variables name

```

For example:

```
user@host# edit policies group tiered_aaa list l1 rule r1 template-activation ta  
variables upstreamBandwidth
```

5. (Optional) Configure the value for the variable.

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables name]  
user@host# set value value
```

For example:

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables  
upstreamBandwidth]  
user@host# set value rateParameter
```

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

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables name]  
user@host# set type type
```

For example:

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables  
upstreamBandwidth]  
user@host# set type rate
```

7. (Optional) Verify the template activation action configuration.

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta]  
user@host# show
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Policy Information Model on page 62
 - Overview of Global and Local Parameters on page 81

Configuring Traffic-Class Actions (SRC CLI)

Use this action to put packets in a particular traffic class. You can configure traffic-class actions for JUNOS policy rules.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statement to configure traffic-class actions:

```
policies group name list name rule name traffic-class name {  
    traffic-class traffic-class ;  
    description description ;  
}
```

To configure a traffic-class action:

1. From configuration mode, enter the traffic-class configuration. For example, in this procedure, *tca* is the name of the traffic-class action.

```
user@host# edit policies group junose list class rule pr traffic-class tca
```

2. (Optional) Configure the name of the traffic-class profile that is applied to a packet when it passes through the router.

```
[edit policies group junose list class rule pr traffic-class tca]
user@host# set traffic-class traffic-class
```

3. (Optional) Enter a description for the traffic-class action.

```
[edit policies group junose list class rule pr traffic-class tca]
user@host# set description description
```

4. (Optional) Verify the traffic-class action configuration.

```
[edit policies group junose list class rule pr traffic-class tca]
user@host# show
traffic-class TCent;
description "Traffic class action";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Traffic-Mirror Actions (SRC CLI) on page 190
 - Configuring Traffic-Shape Actions (SRC CLI) on page 191
 - Configuring Traffic-Class Actions (C-Web Interface)

Configuring Traffic-Mirror Actions (SRC CLI)

Use this action to mirror traffic from a destination to a source or from a source to a destination. You can configure traffic-mirror actions for JUNOS filter input policy rules.

Before you use traffic-mirror actions, configure forwarding options on JUNOS routing platforms for port mirroring and next-hop group. For information about how these features work on the router, see the JUNOS Policy Framework Configuration Guide.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

The rule containing a traffic-mirror action must comply with these conditions:

- It must be combined with forward actions in the same rule. One of the forward actions must accept the traffic if the source and/or destination IP addresses do not match the conditions.
- It contains either no classify-traffic condition or only one classify-traffic condition.

- It can be marked for accounting.

Use the following configuration statement to configure a traffic-mirror action:

```
policies group name list name rule name traffic-mirror name {
  description description ;
}
```

To configure a traffic-mirror action:

1. From configuration mode, enter the traffic-mirror configuration. For example, in this procedure, fromSubnets is the name of the traffic-mirror action.

```
user@host# edit policies group junos list mirror rule pr traffic-mirror fromSubnets
```

2. (Optional) Enter a description for the traffic-mirror action.

```
[edit policies group junos list mirror rule pr traffic-mirror fromSubnets]
user@host# set description description
```

3. (Optional) Verify the traffic-mirror action configuration.

```
[edit policies group junos list mirror rule pr traffic-mirror fromSubnets]
user@host# show
description "Traffic mirroring action for subnet.";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Classify-Traffic Conditions on page 113
 - Configuring Forward Actions (SRC CLI) on page 159
 - Configuring Traffic-Mirror Actions (C-Web Interface)

Configuring Traffic-Shape Actions (SRC CLI)

Traffic-shape actions specify the maximum rate of traffic transmitted on an interface. You can create traffic-shape actions in JUNOS shaping policy rules.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statements to configure traffic-shape actions:

```
policies group name list name rule name traffic-shape name {
  rate rate ;
  description description ;
}
```

To configure a traffic-shape action:

1. From configuration mode, enter the traffic-shape configuration. For example, in this procedure, tsa is the name of the traffic-shape action.

```
user@host# edit policies group junos list trafficShaping rule shaping traffic-shape tsa
```

2. (Optional) Configure the maximum transmission rate.

```
[edit policies group junos list trafficShaping rule shaping traffic-shape tsa]
user@host# set rate rate
```

3. (Optional) Enter a description for the traffic-shape action.

```
[edit policies group junos list trafficShaping rule shaping traffic-shape tsa]
user@host# set description description
```

4. (Optional) Verify the traffic-shape action configuration.

```
[edit policies group junos list trafficShaping rule shaping traffic-shape tsa]
user@host# show
rate 10200000;
description "Traffic-shaping action";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Configuring Traffic-Class Actions (SRC CLI) on page 189
 - Configuring Traffic-Mirror Actions (SRC CLI) on page 190
 - Configuring Traffic-Shape Actions (C-Web Interface)

Configuring User Packet Class Actions (SRC CLI)

Use this action to put packets in a particular user packet class. You can configure user packet class actions for JUNOS policy rules.

The type of action that you can create depends on the type of policy rule. See “Policy Information Model” on page 62.

Use the following configuration statement to configure user packet class actions:

```
policies group name list name rule name user-packet-class name {
  user-packet-class user-packet-class ;
  description description ;
}
```

To configure a user packet class action:

1. From configuration mode, enter the user packet class configuration. For example, in this procedure, upca is the name of the user packet class action.

```
user@host# edit policies group junos list class rule pr user-packet-class upca
```

2. (Optional) Configure the user packet class that is applied to a packet when it passes through the router.

```
[edit policies group junose list class rule pr user-packet-class upca]
user@host# set user-packet-class user-packet-class
```

3. (Optional) Enter a description for the user packet class action.

```
[edit policies group junose list class rule pr user-packet-class upca]
user@host# set description description
```

4. (Optional) Verify the user packet class action configuration.

```
[edit policies group junose list class rule pr user-packet-class upca]
user@host# show
user-packet-class 5;
description "User packet class action";
```

- Related Topics**
- Before You Configure SRC Policies on page 105
 - Delivering QoS Services in a Cable Environment on page 73
 - Configuring JUNOS Secondary Input Policy Conditions (SRC CLI) on page 142

Chapter 8

Policy Examples (SRC CLI)

- Example: Creating Access Policies for Subscribers on page 195
- Example: Providing Tiered Internet Services with Policing on page 199
- Example: Providing Premium Services on page 203

Example: Creating Access Policies for Subscribers

In this example, the service provider manages an interface on the router. The interface is associated with a subscriber. The access policy is a default policy that supports various types of subscribers and interfaces. Some examples are DHCP, static IP subscribers, and PPP subscribers.

From the service provider's perspective, the service provider's network is on the inside, and the enterprise network is on the outside. Ingress traffic flows from the enterprise network to the service provider's network. Egress traffic flows from the service provider's network to the enterprise network.

The default policy installed on the interface sets the context of other services that the subscriber will activate later. The default policy can restrict subscriber access to the network or provide a default access. You can also use the default policy to create a walled garden effect by sending subscribers to the SAE server and requiring them to activate a service before they can access other services in the system. (The term walled garden is used to describe an environment in which a service provider limits a subscriber's access to Web content and services.)

The precedence of the policy rules in default policies is very important. When the related service is activated, the service policy needs a high priority (low value) so that the service policy is used instead of the default policy.

Types of Policies

The policy used for access depends on the type of services that it will be used for. Generally, policies with filter, forward, rate-limit or policer, and next-hop actions are used.

Sample Access Policies

This section contains examples of access policies for DHCP subscribers and PPP subscribers. In both of these examples, there are two content providers. Traffic destined for the content provider networks is sent to the residential portal by means

of a next-hop action that forwards traffic to the virtual IP address of the portal. (See *SRC-PE Sample Applications Guide*.)

Traffic to the portal has a high priority and is not affected by other service policies. This way, the subscriber can always access the portal. Traffic from the network is forwarded without any restrictions.

DHCP Policy Group

The following information shows the configuration details of the DHCP policy group.

Policy List Out

```
[edit policies folder sample folder junose group DHCP list out]
user@host# show
role junose-ipv4;
applicability output;
rule forward {
    type junose-ipv4;
    precedence 500;
    forward forward {
    }
    traffic-condition any {
    }
}
```

Policy List In

```
[edit policies folder sample folder junose group DHCP list in]
user@host# show
role junose-ipv4;
applicability input;
rule forward-to-SSP {
    type junose-ipv4;
    precedence 200;
    forward forward {
    }
    traffic-condition ssp {
        destination-network {
            network {
                ip-address virtual_ipAddress;
                ip-mask 255.255.255.255;
                ip-operation 1;
            }
        }
    }
}
rule forward-cl-dhcp {
    type junose-ipv4;
    precedence 200;
    forward Fo {
    }
    traffic-condition cl-dhcp {
        protocol-port-condition {
            protocol udp;
            protocol-operation is;
        }
    }
}
```

```

        ip-flags 0;
        ip-flags-mask 0;
        destination-port {
            port {
                port-operation eq;
                from-port 67;
            }
        }
        source-port {
            port {
                port-operation neq;
            }
        }
    }
}

rule cp-to-ssp {
    type junose-ipv4;
    precedence 500;
    next-hop to-ssp {
        next-hop-address virtual_ipAddress;
    }
    traffic-condition content-provider-network-1 {
        destination-network {
            network {
                ip-address 10.10.40.0;
                ip-mask 255.255.255.0;
                ip-operation 1;
            }
        }
    }
    traffic-condition content-provider-network-2 {
        destination-network {
            network {
                ip-address 172.16.0.0;
                ip-mask 255.255.0.0;
                ip-operation 1;
            }
        }
    }
}

```

PPP Policy Group

The following information shows the configuration details of the PPP policy group.

Policy List Out

```

[edit policies folder sample folder junose group PPP list out]
user@host# show
role junose-ipv4;
applicability output;
rule forward {
    type junose-ipv4;
    precedence 500;
    forward forward {
    }
}

```

```

    traffic-condition any {
    }
}

```

Policy List In

```

[edit policies folder sample folder junose group PPP list in]
user@host# show
role junose-ipv4;
applicability input;
rule forward-to-SAE {
    type junose-ipv4;
    precedence 200;
    forward forward {
    }
    traffic-condition sae {
        destination-network {
            network {
                ip-address virtual_ipAddress;
                ip-mask 255.255.255.255;
                ip-operation 1;
            }
        }
    }
}
rule cp-to-ssp {
    type junose-ipv4;
    precedence 500;
    next-hop to-ssp {
        next-hop-address virtual_ipAddress;
    }
    traffic-condition content-provider-network-1 {
        destination-network {
            network {
                ip-address 10.10.40.0;
                ip-mask 255.255.255.0;
                ip-operation 1;
            }
        }
    }
    traffic-condition content-provider-network-2 {
        destination-network {
            network {
                ip-address 172.16.0.0;
                ip-mask 255.255.0.0;
                ip-operation 1;
            }
        }
    }
}
}

```

- Related Topics**
- Policy Information Model on page 62
 - Policy Components on page 60
 - Before You Configure SRC Policies on page 105

Example: Providing Tiered Internet Services with Policing

In this scenario, the service provider offers three tiered Internet services to its subscribers:

- Gold, which provides a bandwidth of up to 5 Mbps.
- Silver, which provides a bandwidth of up to 1 Mbps.
- Bronze, which provides a bandwidth of up to 64 Kbps.

One of the tiered Internet services controls the traffic at a given time. Accounting data is collected for the tiered services.

A default policy is needed to establish the context of the tiered service. The subscriber has an IP interface in the network; the access point has a default policy that prevents the subscriber from using a tiered Internet service until the service is activated.

From the service provider's perspective, the service provider's network is on the inside, and the enterprise network is on the outside. Ingress traffic flows from the enterprise network to the service provider's network. Egress traffic flows from the service provider's network to the enterprise network.

Types of Policies

JUNOS policies use the rate-limit action to control bandwidth, and JUNOS policies use the policer action to control bandwidth. You can also use QoS conditions and scheduler actions to provide tiered Internet services.

Sample JUNOS Rate-Limiting Policy

The sample JUNOS policy has a local parameter `bw`, which is used in the rate-limit action both on input and output directions.

In this example, the committed action is forward, whereas the conformed and exceeded actions are set to filter.

The following information shows the configuration details of the Internet tiered policy group for JUNOS routers.

Local Parameter

```
[edit policies folder sample folder common group internet-tiered local-parameters]
user@host# show
parameter bw {
    default-value 5000000;
    type rate;
}
```

Policy List je-out

```
[edit policies folder sample folder common group internet-tiered list je-out]
user@host# show
role junose-ipv4;
applicability output;
rule the-limit {
  type junose-ipv4;
  precedence 600;
  accounting;
  rate-limit limit {
    committed-action {
      forward {
      }
    }
    conformed-action {
      filter {
      }
    }
    exceed-action {
      filter {
      }
    }
  }
  type two_rate;
  committed-rate bw;
  committed-burst 500000;
  peak-rate bw;
  peak-burst 500000;
}
traffic-condition any {
}
}
```

Policy List je-in

```
[edit policies folder sample folder common group internet-tiered list je-in]
user@host# show
role junose-ipv4;
applicability input;
rule the-limit {
  type junose-ipv4;
  precedence 600;
  accounting;
  rate-limit limit {
    committed-action {
      forward {
      }
    }
    conformed-action {
      filter {
      }
    }
    exceed-action {
      filter {
      }
    }
  }
  type two_rate;
  committed-rate bw;
```

```

        committed-burst 500000;
        peak-rate bw;
        peak-burst 500000;
    }
    traffic-condition any {
    }
}

```

Sample JUNOS Policer Policy

The sample JUNOS policy has a local parameter `bw`, which is used in the policer action both on input and output directions.

In this example, packets that exceed the bandwidth limit are filtered.

The following information shows the configuration details of the Internet tiered policy group for JUNOS routing platforms.

Local Parameter

```

[edit policies folder sample folder common group internet-tiered local-parameters]
user@host# show
parameter bw {
    default-value 5000000;
    type rate;
}

```

PolicyList j-out

```

[edit policies folder sample folder common group internet-tiered list j-out]
user@host# show
role junos;
applicability output;
rule PR {
    type junos-filter;
    precedence 100;
    policer PA {
        packet-action packet0 {
            filter {
            }
        }
        bandwidth-limit bw;
        bandwidth-limit-unit bps;
        burst 15000;
    }
}

```

PolicyList j-in

```

[edit policies folder sample folder common group internet-tiered list j-in]
user@host# show
role junos;
applicability input;

```

```

rule PR {
  type junos-filter;
  precedence 100;
  policer PA {
    packet-action packet0 {
      filter {
      }
    }
    bandwidth-limit bw;
    bandwidth-limit-unit bps;
    burst 15000;
  }
}

```

Defining the Tiered Internet Services

You need to create three SAE services—Gold, Silver, and Bronze.

Assign to the new service one of the Internet-tiered policy groups that we created in the last section.

For each service, define a substitution value for the bw parameter. For the Gold service, the bw value is 5 Mbps; for the Silver service, the bw value is 1 Mbps; and for the Bronze service, the bw value is 64 Kbps.

Internet-Gold Service

```

[edit services global service Internet-Gold]
user@host# show
description "Example for rate limited internet (requires matching default
policies)";
type normal;
category Internet;
policy-group /sample/common/internet-tiered;
radius-class Internet-Gold;
status active;
parameter {
  substitution "bw = 5000000";
}

```

Internet-Silver Service

```

[edit services global service Internet-Silver]
user@host# show
description "Example for rate limited internet (requires matching default
policies)";
type normal;
category Internet;
policy-group /sample/common/internet-tiered;
radius-class Internet-Silver;
status active;
parameter {
  substitution "bw = 1000000";
}

```

Internet-Bronze Service

```
[edit services global service Internet-Bronze]
user@host# show
description "Example for rate limited internet (requires matching default
policies)";
type normal;
category Internet;
policy-group /sample/common/internet-tiered;
radius-class Internet-Bronze;
status active;
parameter {
    substitution "bw = 64000";
}
```

- Related Topics**
- Policy Information Model on page 62
 - Policy Management Overview on page 59
 - Policy Components on page 60
 - Before You Configure SRC Policies on page 105

Example: Providing Premium Services

This scenario shows how service providers can offer premium services, such as video on demand, video conferencing, and voice over IP (VoIP). These types of services are turned on for short periods of time while the premium service is being used.

From the service provider's perspective, the service provider's network is on the inside, and the enterprise network is on the outside. Ingress traffic flows from the enterprise network to the service provider's network. Egress traffic flows from the service provider's network to the enterprise network.

In this example, two content providers provide premium services. One provides a music service, and the other provides a news service.

Types of Policies

The policy used for premium services depends on the type of service being used. Generally, policies with filter, forward, rate-limit or policer actions, and QoS features are used.

The policy rules in premium services typically have a higher priority (smaller precedence number) than other services and default policies. In this case, the policy rules in the content provider service policies have a priority of 400. The default policy rule has a priority of 500.

The default policy uses the next-hop action to send all traffic destined for the networks of these content providers to the portal (see "Example: Creating Access Policies for Subscribers" on page 195). When the content provider service is activated, the forward action is taken for packets destined for the content provider network.

Sample JUNOS and JUNOSe Content Provider Policies

The sample content provider policy group includes policy lists for both JUNOS and JUNOSe policies. The following information shows the configuration details of the premium service policy group.

```
policyGroupName=content-provider,ou=common,ou=sample,o=Policies,o=umc
```

PolicyList je-out

```
[edit policies folder sample folder common group content-provider list je-out]
user@host# show
role junose-ipv4;
applicability output;
rule from-content-provider {
    type junose-ipv4;
    precedence 400;
    accounting;
    forward forward {
    }
    traffic-condition content-provider {
        source-network {
            network {
                ip-address service_ipAddress;
                ip-mask service_ipMask;
                ip-operation 1;
            }
        }
    }
}
```

PolicyList j-out

```
[edit policies folder sample folder common group content-provider list j-out]
user@host# show
role junos;
applicability output;
rule PR {
    type junos-filter;
    precedence 100;
    forward FA {
    }
    traffic-condition content-provider {
        source-network {
            network {
                ip-address service_ipAddress;
                ip-mask service_ipMask;
                ip-operation is;
            }
        }
    }
}
```

PolicyList je-in

```
[edit policies folder sample folder common group content-provider list je-in]
user@host# show
role junose-ipv4;
applicability input;
rule to-content-provider {
    type junose-ipv4;
    precedence 400;
    accounting;
    forward forward {
    }
    traffic-condition content-provider {
        destination-network {
            network {
                ip-address service_ipAddress;
                ip-mask service_ipMask;
                ip-operation 1;
            }
        }
    }
}
```

PolicyList j-in

```
[edit policies folder sample folder common group content-provider list j-in]
user@host# show
role junos;
applicability input;
rule PR {
    type junos-filter;
    precedence 100;
    forward FA {
    }
    traffic-condition content-provider {
        destination-network {
            network {
                ip-address service_ipAddress;
                ip-mask service_ipMask;
                ip-operation is;
            }
        }
    }
}
```

Defining the Premium Services

You need to create two SAE services—one for the news service and one for the music service. Assign to the new service the content-provider policy group that we created in the last section.

For each service, define a substitution value for the `service_ipAddress` and `service_ipMask` parameters. Note that each content provider has a different `service_ipAddress` parameter.

Music Service

The music service is provided by the XYZ company, which is a content provider.

```
[edit services global sae-service Music]
user@host# show
type normal;
policy-group /sample/content-provider;
status active;
available;
parameter {
    service-ip-address 10.20.30.0;
    service-ip-mask 255.255.255.0;
}
```

News Service

The news service is provided by the ABC company, which is a content provider.

```
[edit services global sae-service News]
user@host# show
description "Example for content-provider in different network";
type normal;
category News;
url http://the.news.com;
policy-group /sample/common/content-provider;
radius-class News;
status active;
parameter {
    service-ip-address 10.20.40.0;
    service-ip-mask 255.255.255.0;
}
```

- Related Topics**
- Delivering QoS Services in a Cable Environment on page 73
 - Policy Information Model on page 62
 - Policy Management Overview on page 59
 - Before You Configure SRC Policies on page 105

Part 3

Generating Policies by Specifying Parameters

- Defining and Acquiring Values for Parameters on page 209

Chapter 9

Defining and Acquiring Values for Parameters

- Parameters and Substitutions on page 209
- Value Acquisition for Single Subscriptions on page 210
- Value Acquisition for Multiple Subscriptions on page 211
- Defining Parameters for the SRC Software on page 212
- Formatting Substitutions on page 214
- Parameter Names and Types on page 215
- Expressions in Parameters on page 215
- Adding Comments to Substitutions on page 223
- Validating Substitutions on page 224
- Example: Parameter Value Substitution on page 224

Parameters and Substitutions

Each subscriber who uses the SRC network is associated with an entry in the directory. You do not need to configure a policy for each subscriber, however. You can define a smaller number of policies that contain *parameters*. A parameter is a general definition for a property, such as an IP address, and is analogous to a variable in a computer program.

The SRC software defines some global parameters and system (runtime) parameters in the policy repository. You can also define your own global parameters in the policy repository, your own local parameters in policy groups, and your own local parameters in other specified items, such as services. When the SAE activates a subscription to a service for a subscriber, it constructs an exact policy for that subscriber by obtaining specific values for parameters. The SAE acquires one or more values for each parameter from a number of different sources. These sources can also contain local parameters for which other sources can provide specific values. The SAE selects a value based on a ranking of sources from specific to general. The process of providing a value or a new definition for a parameter is a *substitution*.

One or more sources can define a parameter as fixed. Fixing prevents acquisition of values from more specific sources in the ranking list. For example, if a parameter is fixed in a subscription for a parent subscriber, a subordinate subscriber cannot provide a more specific value for a parameter in the subscription it inherits from the

parent. If a parameter is fixed in more than one place, the SAE uses the setting in the source that is classified as more general.

You can fix a parameter without specifying a value. Doing so specifies that the value for the parameter cannot come from a more general source than the one that contains the fixed setting and that a value will be available at some point. For example, you could fix the value of the system parameter `interface_speed` in the service scope to prevent more specific sources in the ranking list, such as subscribers, from providing a value for this parameter. The SAE could acquire an actual value for this parameter when it starts managing an interface.

The SAE fixes global and system parameters at a set point in the acquisition chain. Consequently, the SAE can acquire values for these types of parameters only from a service scope, from information the SAE obtains when it starts managing an interface, or from the default value in the global parameter definition.

When you are designing policies, services, portals, and applications, you need to consider how you will use substitutions throughout the software. As a simple example, you can define the general settings for a rate limiter in a policy, insert a parameter for a rate in the policy, and provide specific values for the rate in each service that uses this policy. In a more complex example, you can use parameters and substitutions to track the use of a particular service by different departments in an enterprise.

- Related Topics**
- Overview of Global and Local Parameters on page 81
 - Parameter Types on page 82
 - Expressions in Parameters on page 215
 - Formatting Substitutions on page 214
 - Example: Parameter Value Substitution on page 224

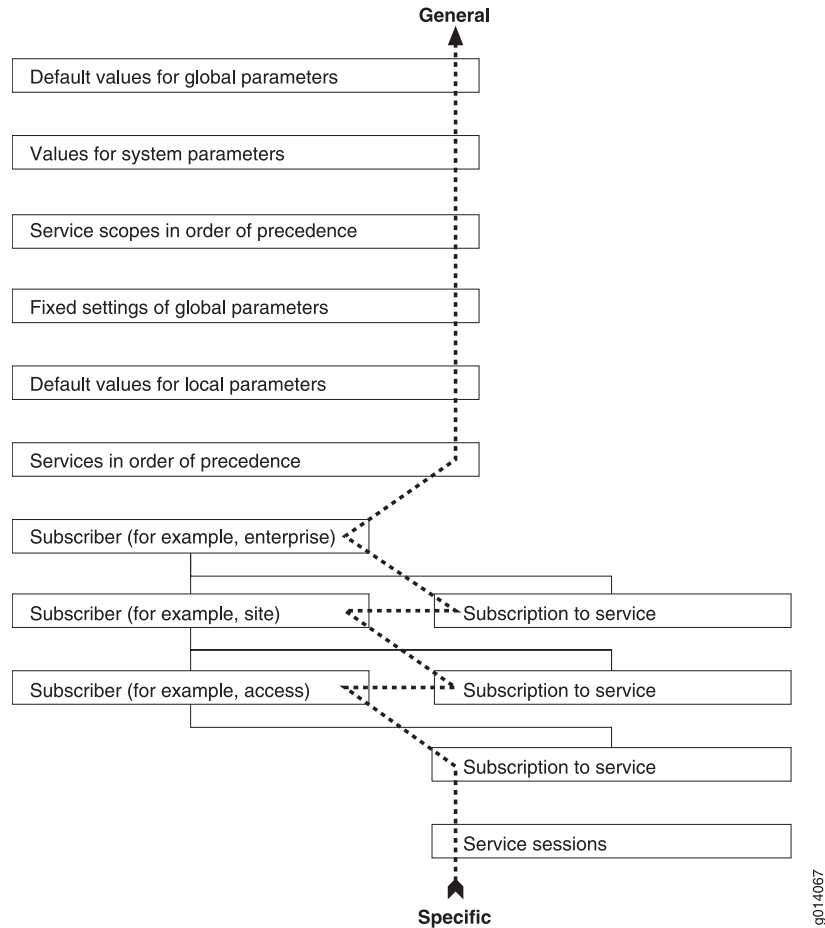
Value Acquisition for Single Subscriptions

When a subscriber has a single subscription to a service, the SAE ranks sources in the following order when it selects values for parameters:

1. The service sessions associated with the subscriber
2. The subscriber's subscription to a service and then the subscriber
3. Each parent subscriber's subscription and then the parent subscriber
4. The services in order of the precedences defined for their associated service scopes
5. The default values for the local parameters in the policy group
6. Fixed settings of all global parameters defined in the policy repository
7. The service scopes, in order of precedence for each of the services. See:
8. Values for system parameters that are available only when the SAE starts managing the interface (for example, actual bandwidth rates)
9. The default values for global parameters defined in the policy repository

Figure 12 on page 211 illustrates how the SAE selects values for a subscriber with one subscription to a service.

Figure 12: Value Acquisition for Single Subscriptions



- Related Topics**
- Parameters and Substitutions on page 209
 - Value Acquisition for Multiple Subscriptions on page 211
 - Overview of Restricting and Customizing Services for Subscribers on page 28
 - Example: Parameter Value Substitution on page 224

Value Acquisition for Multiple Subscriptions

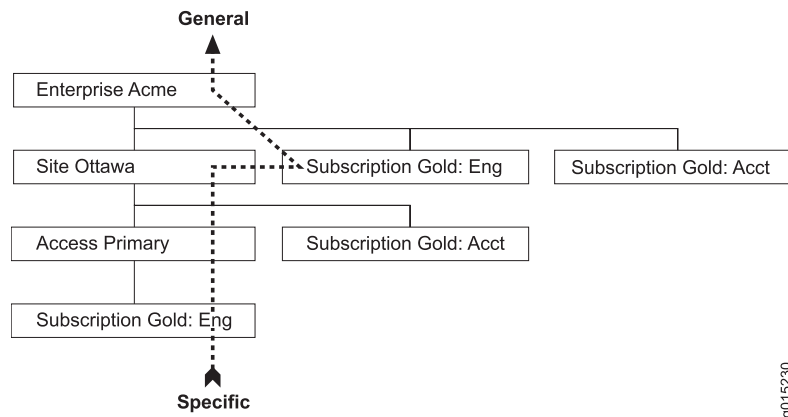
A subscriber can have multiple subscriptions, each with different service parameters, to the same service. When a subscriber has multiple subscriptions to the same service, each subscription has a different name. The name is determined by the parameters. Different subscribers can have subscriptions with the same name.

As described previously, the SAE considers the subscriptions of parent subscribers when it acquires parameters for the policy of a subordinate subscriber who has one

subscription to a service. When acquiring parameters for the policy of a subordinate subscriber who has multiple subscriptions to a service, however, the SAE considers the parent subscriber's subscription only if it has the same name as the subordinate subscriber's subscription.

Figure 13 on page 212 shows an example that illustrates this concept.

Figure 13: Value Acquisition for Multiple Subscriptions



In this example, an enterprise called Acme contains a site called Ottawa that contains an access called Primary. The access has a subscription called Gold:Eng to the service called Gold; the site has a subscription called Gold:Acct to the same service; and the enterprise has two subscriptions, Gold:Eng and Gold:Acct, to the service.

When the IT manager activates the Gold:Eng subscription for the access, the SAE will consider the parameters in the subscriptions Gold:Eng for the access and the enterprise; however, the SAE will not consider the parameters in the subscriptions called Gold:Acct for the site or the enterprise.

The SAE acquires parameters from other sources in the same way whether the subscriber has multiple subscriptions to a service or a single subscription to a service.

Related Topics

- Parameters and Substitutions on page 209
- Value Acquisition for Single Subscriptions on page 210
- Overview of Restricting and Customizing Services for Subscribers on page 28
- Example: Parameter Value Substitution on page 224

Defining Parameters for the SRC Software

You can define parameters for different items in the SRC software. Depending on the item, you can define parameters with the SRC CLI, the C-Web interface, and with LDAP clients, with SRC applications, or with other applications through SRC APIs.

Table 13 on page 213 shows the items for which you can define parameters and the methods you can use to define parameters for these items. See the documentation specified in the table for information about how to define parameters for each item.

Table 13: Parameter Definitions

Items That Contain Parameter Definitions	Methods by Which You Can Define the Parameter	Information That Describes How to Define Parameters
Global parameters, which you define in the runtime parameters in the policy repository folder.	SRC CLI C-Web interface	<i>“Overview of Global and Local Parameters” on page 81</i>
Local parameters, which you define in policy groups.	SRC CLI C-Web interface	<i>“Overview of Global and Local Parameters” on page 81</i>
System parameters, which are contained in the runtime parameters folder in the policy repository folder.	Subscriber sessions SRC network-values obtained when the SAE starts managing an interface	<i>“Parameter Types” on page 82</i>
Scopes in order of the precedence	SRC CLI C-Web interface	<i>“Setting Parameter Values for Services (SRC CLI)” on page 8</i> <i>“Overview of Restricting Simultaneous Activation of Services” on page 26</i>
Services in order of the precedence associated with the scopes associated with the service	SRC CLI C-Web interface	<i>“Setting Parameter Values for Services (SRC CLI)” on page 8</i> <i>“Overview of Restricting and Customizing Services for Subscribers” on page 28</i>
Subscribers	SRC CLI C-Web interface LDAP client if an external directory contains subscriber data Enterprise service portals	Overview of Configuring Subscribers and Subscriptions Adding Subscribers (SRC CLI) Adding Subscribers (C-Web Interface)

Table 13: Parameter Definitions (continued)

Items That Contain Parameter Definitions	Methods by Which You Can Define the Parameter	Information That Describes How to Define Parameters
Subscriptions	SRC CLI	Configuring Subscriptions (SRC CLI)
	C-Web interface	Configuring TCP Conditions (C-Web Interface)
	Residential portal or enterprise service portal	<i>SRC-PE Sample Applications Guide</i> Overview of Dynamic Service Activator
	Dynamic Service Activator	SAE CORBA remote API documentation on the Juniper Networks Web site at http://www.juniper.net/techpubs/software/management/src/api-index.html
	SAE's CORBA remote API	
Sessions	Residential portal	<i>SRC-PE Sample Applications Guide</i> Overview of Dynamic Service Activator
	Dynamic Service Activator	SAE CORBA remote API documentation on the Juniper Networks Web site at http://www.juniper.net/techpubs/software/management/src/api-index.html
	SAE's CORBA remote interface	

- Related Topics**
- Parameters and Substitutions on page 209
 - Overview of Global and Local Parameters on page 81
 - Parameter Types on page 82
 - Example: Parameter Value Substitution on page 224

Formatting Substitutions

Some SRC components handle the substitution syntax for you. For example, in policy configuration you to enter settings in fields, and it formats these settings in the correct syntax. In addition, IT managers or residential subscribers can enter settings through portals, and the portal formats these items in the correct syntax. You enter some substitutions (in the correct syntax) in SRC CLI. If you develop a portal that uses substitutions, use the correct syntax in the code for that portal.

A substitution has the following syntax:

```
[ ! ]<parameterName>[ :<role>]*=[<expression>]
[ //<comment> ]
```


- `!`—Fixes the substitution
- `< parameterName >` —Name of the parameter; either a name that you define or a name that is specified by the SRC software. Parameter names are case sensitive. If you are defining a substitution for a global parameter, make sure that the case of the parameter name in the substitution matches the case of the global parameter.
- `< role >` —Category, or type, of the parameter. In the software the terms role and type are used interchangeably.
- `< expression >` —A definition for the parameter
- `// < comment >` —A comment about a substitution that appears on a new line after the substitution syntax

- Related Topics**
- Parameters and Substitutions on page 209
 - Parameter Names and Types on page 215
 - Expressions in Parameters on page 215
 - Adding Comments to Substitutions on page 223

Parameter Names and Types

Parameter names identify the local or global parameters that are defined by you or that are specified by the SRC software. The parameter name is a string of alphanumeric characters starting with a letter that does not contain spaces or special characters.

Parameters fall into different categories, known as types. For example, a parameter that defines an IP address has the type address.

- Related Topics**
- Parameters and Substitutions on page 209
 - Parameter Types on page 82
 - Expressions in Parameters on page 215
 - Defining Parameters for the SRC Software on page 212

Expressions in Parameters

An expression in a parameter definition can take one of the following values:

- An explicit value; for example, 1000000
- Another parameter; for example, a parameter called bodDestPort
- A mathematical expression that can include a combination of:
 - Parameters
 - Numbers—Integers and floating point numbers
 - Strings

- IPv4 addresses
- Ranges of numbers, strings, and addresses
- Lists of values, such as lists of protocols
- Maps—List of pairs of attributes and corresponding values
- One keyword, **not**
- Separators
- Operators

For example, $x = 1 ? \text{rate} : 2 * \text{rate}$

The syntax for mathematical expressions is based primarily on Java syntax, although a few items use a proprietary syntax. When evaluating mathematical expressions, the SRC software:

- Follows a defined order for the precedence of operators (see “Expressions in Parameters” on page 215).
- Performs all evaluations in long integer format until it finds an argument or result that is in Java floating point number format. Subsequently, the software performs evaluations in Java double floating point number format.
- Evaluates only subordinate expressions that meet the conditions for evaluation.
 - Evaluates only subordinate expressions that contain numbers and not parameters.
 - Stops the evaluation and substitutes the partial evaluation if an argument in double floating number format becomes an argument to an operator that takes only integers.
- Behaves in the same way as a Java evaluation if intermediate evaluations exceed or fall below the long integer range or the double floating point number range.
- Follows the Java rules for raising exceptions. For example, the software raises an exception if:
 - An evaluation involves a division by zero.
 - Literal numbers exceed the long integer limit or the double floating point number limit.

The following sections describe how to format the items that you can use in an expression.

Specifying Parameter Names

Observe the following rules when you are specifying parameter names:

- Enter a string of alphanumeric characters starting with a letter.
- Do not use spaces or special characters. For example, do not use the at sign (@) in a parameter name.

- You can use the underscore (_) and the dollar sign (\$). Use the dollar sign to encode special characters by entering the Unicode equivalent of the character in hexadecimal format after the dollar sign. For example, use \$0040 to encode the at sign (@).

Formatting Numbers

Observe the following rules when you are formatting numbers:

- Enter a digit after the decimal point in a floating point number. For example, you can use the number 4.0, but not the number 4.
- Do not enter characters that specify the type of number after that number. For example, do not enter the character L after a number to indicate that the number is a long integer.

Formatting Strings

Use Java syntax for strings; enclose strings in double quotation marks.

Example—" engineering"

Observe the following rules when you are formatting strings:

- Do not use octal escape sequences in strings. For example, do not use the escape sequence \137 in a string.
- Do not use Unicode escape sequences. For example, do not use the escape sequence \u80A6 in a string.

Using IPv4 Addresses

Use the following format for IP addresses:

`<string>.<string>.<string>.<string> | '<string>.<string>.<string>.<string>'`
`<string>` is a set of digits in the range 0–255
 Example—'192.0.2.1'

Single quotation marks around an item indicate that it represents an address; however, for IPv4 addresses, the quotation marks are optional.

Specifying Ranges

To specify a range of numbers, strings, and addresses, use two dots between the arguments.

Example—192.0.2.1..192.0.3.1

Formatting Lists

To specify a list of values, enclose a set of subordinate expressions separated by commas in a pair of square brackets.

Example—[ip, icmp, ftp]

Formatting Maps

Maps are used to specify values that have optional and interdependent attributes. For example, when you define an application object through the Enterprise Manager portal, you can select a number of attributes and specify particular values for them. Depending on the value of the attribute, other attributes are possible or required.

To format a map, specify a list of pairs of attributes and corresponding values. Separate the pairs with commas, and enclose the list in curly brackets (braces).

Example—{applicationProtocol="ftp" , sourcePort=123, inactivityTimeout=60}

Using Keywords

The SRC software ignores all Java keywords in substitutions, so that you can use Java keywords for identifiers such as variable names, function names, and attribute names in maps. The SRC software accepts one keyword, **not**, which is used to indicate conditions that do not match a specified value. For more information about the **not** keyword, see “Expressions in Parameters” on page 215.

Using Separators

You cannot use a dot (.) as a separator. You can use other Java separators in the ways that Java supports.

Using Operators

Table 14 on page 218 shows the operations and corresponding operators that the SRC software supports for substitutions. Most of the operators are Java operators, although a few operators are proprietary. You cannot use Java operators that do not appear in this table.

Table 14: Operations That You Can Use in Expressions

Operation	Operator	Number of Arguments	Result If Different from Java Conventions	Conditions for Evaluation	Example
Bitwise AND of the arguments	&	Two		Both arguments must be integers	234567 & 876543
Bitwise exclusive OR of the arguments	^	Two		Both arguments must be integers	234567 ^ 876543

Table 14: Operations That You Can Use in Expressions (continued)

Operation	Operator	Number of Arguments	Result If Different from Java Conventions	Conditions for Evaluation	Example
Bitwise inclusive OR of the arguments		Two		Both arguments must be integers	234567 876543
Bitwise negation of the argument	~	One		Argument must be an integer	~234567
Difference between two arguments	-	Two		Both arguments must be numbers	876543 - 234567
Division of the first argument by the second argument	/	Two	Result of operation in double format	Both arguments must be numbers	589 / 756
Equal	= =	Two	Nonzero number if the arguments are equal	Both arguments must be numbers	rate = = 5
Greater than	>	Two	Nonzero integer if the first argument is greater than the second argument	Both arguments must be numbers	rate > 5
Greater than or equal to	> =	Two	Nonzero integer if the first argument is greater than or equal to the second argument	Both arguments must be numbers	rate > = 5

Table 14: Operations That You Can Use in Expressions *(continued)*

Operation	Operator	Number of Arguments	Result If Different from Java Conventions	Conditions for Evaluation	Example
If... then... else... operation	?:	Three	If the first argument is nonzero, then the result is the second argument, else the result is the third argument	First argument must be a number	"x == 1 ? rate : 2*rate"
Less than	<	Two	Nonzero integer if the first argument is less than the second argument	Both arguments must be numbers	rate < 5
Less than or equal to	< =	Two	Nonzero integer if the first argument is less than or equal to the second argument	Both arguments must be numbers	rate < = 5
Logical AND	&&	Two	Nonzero integer if both the arguments are nonzero	Both arguments must be numbers	x == 1 && y > = 5
Logical NOT	!()	One	Zero if the argument is nonzero	All arguments must be numbers	! x == y
Logical OR		Two	Nonzero integer if at least one of the arguments is nonzero	Both arguments must be numbers	x == 1 y > = 5
Maximum of the arguments, max() = -infinity	max()	Zero or more		All arguments must be numbers	max (1, 3, 2, 4)

Table 14: Operations That You Can Use in Expressions (continued)

Operation	Operator	Number of Arguments	Result If Different from Java Conventions	Conditions for Evaluation	Example
Minimum of the arguments, <code>min()</code> = + infinity	<code>min()</code>	Zero or more		All arguments must be numbers	<code>min (1, 3, 2, 4)</code>
Negation	<code>-</code>	One		Argument must be a number	<code>-5</code>
Not equal	<code>!=</code>	Two	Nonzero integer if the arguments are not equal	Both arguments must be numbers	<code>rate != 5</code>
Not match	<code>not</code>	One		None – expressions with this operator cannot be evaluated	<code>not 192.0.2.1</code>
Product of the arguments	<code>*</code>	Two		Both arguments must be numbers	<code>rate*2</code>
Raise the first argument to the power of the second argument	<code>**</code>	Two		Both arguments must be numbers	<code>2**16</code>
Range from the first argument to the second argument	<code>..</code>	Two		None—expressions with this operator cannot be evaluated	<code>0..49</code>
Remainder of division of the first argument by the second argument	<code>%</code>	Two		Both arguments must be integers	<code>5%2</code>
Round off the argument to the closest number	<code>round()</code>	One	Integer closest to the argument	Argument must be numbers	<code>round(986532.654)</code>

Table 14: Operations That You Can Use in Expressions *(continued)*

Operation	Operator	Number of Arguments	Result If Different from Java Conventions	Conditions for Evaluation	Example
Round the argument down	<code>floor()</code>	One	Biggest integer less than or equal to the argument	Argument must be numbers	<code>floor (986532.654)</code>
Round the argument up	<code>ceiling()</code>	One	Smallest integer greater than or equal to the argument	Argument must be numbers	<code>ceiling (986532.654)</code>
Shift the first argument left by the number of bits in the second argument	<code>< <</code>	Two		Both arguments must be integers	<code>986532 < < 2</code>
Shift the first argument right by the number of bits in the second argument	<code>> ></code>	Two		Both arguments must be integers	<code>986532 > > 2</code>
Sum of the arguments	<code>+</code>	One or two		Both arguments must be numbers	<code>876 + 345</code> <code>+ 855</code>

The precedence of the Java operators is the same as the precedence in Java; if you are unsure of the precedence of the operators, you can use parentheses to ensure that the software evaluates expressions in the desired way. For example, the following logical OR expression does not need parentheses.

```
x==1 || y>=5
```

You can, however, include parentheses as follows:

```
(x==1) || (y>=5)
```

The following list shows the precedence of the operators from lowest precedence to highest precedence:

- not
- ..
- ?:
- ||
- &&
- |
- ^
- &
- = , !=
- < , > , < = , > =
- < < , > >
- + , - (binary)
- * , / , %
- **
- + , - (unary)
- ~ , !

Related Topics

- Parameters and Substitutions on page 209
- Formatting Substitutions on page 214
- Adding Comments to Substitutions on page 223
- Parameter Names and Types on page 215

Adding Comments to Substitutions

To add a comment on the last line of the substitution:

1. Place the Java single-line comment marker (`//`) at the end of the last line of the substitution.
2. Enter the comment.

There is no limit to the length of the comment you can enter. You do not need to use the new line marker in comments. Any text that follows the comment marker, regardless of how many lines the text spans, is treated as part of the comment.

Example—`//This parameter specifies the QoS rate for this service.`

The SRC software supports only the Java single-line comment marker. You cannot use the comment marker for multiple lines or comment markers for other languages.

- Related Topics**
- Parameters and Substitutions on page 209
 - Formatting Substitutions on page 214

Validating Substitutions

You can validate substitutions with the Enterprise portal.

When validating substitutions, the SRC software:

- Checks the syntax of substitutions. For example, if you incorrectly specify a range by using 3 dots between the arguments instead of 2 dots, the SRC software returns an error.
- Does not check the arguments that you specify for an operator. For example, in the expression 192.0.2.16/28 the software recognizes the forward slash (/) as a division operator, but does not check that the arguments are appropriate for division.

This feature allows SRC components, such as the policy engine, to interpret the expression 192.0.2.16/28 as an IP address and mask rather than a division operation.

- Does not check for consistent use of roles in parameters in a chain of substitutions. For example, consider the following situation:
 1. You define in a policy group a local parameter x with the role network and an expression of y (x:network = y).
 2. You define in a service a parameter y with the role rate and a value of 123 (y:rate = 123).

The software will substitute the value of 123 for x, even though 123 is a rate and not an address. Eventually, however, the substitution will cause problems, and a component such as the policy engine or the SAE will reject the value.

- Related Topics**
- Formatting Substitutions on page 214
 - Example: Parameter Value Substitution on page 224

Example: Parameter Value Substitution

Parameters provide general definitions for configuration properties. You can use parameters in the configuration for policies, services, and subscriptions. Users can define the value for a parameter through an enterprise service portal or a residential portal.



NOTE: The SRC sample data includes the configuration used in this example.

This example shows how to use parameters and substitutions in the SRC software.

- Requirements on page 225
- Overview on page 225
- Configuration on page 228

Requirements

This example uses the following hardware and software components:

- SRC software 1.0.0 and greater
- Sample enterprise service portal available with SRC software 1.0.0 and greater
- C-series controller
- JUNOSe router(s)

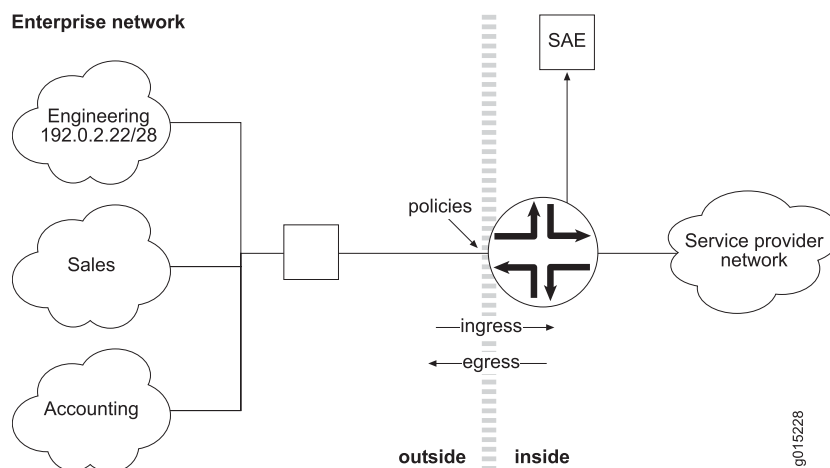
Overview

This configuration has the following characteristics:

- A service that provides a gold-level quality of service
- A department subnet in an enterprise network subscribes to this service with the ability to track and charge the department for the volume of bandwidth used.

Figure 14 on page 225 shows the network in the example.

Figure 14: Network Used in Parameter Substitution Example



From the service provider's perspective, the service provider's network is on the inside, and the enterprise network is on the outside. Ingress traffic flows from the enterprise network to the service provider's network. Egress traffic flows from the service provider's network to the enterprise network. The engineering department

subnet in the enterprise network is the subnet that we will subscribe to the gold-level service and track.

Types of Parameters

The example uses two types of parameters:

- rate—Used to scale the rate limiter
- network—Used to specify IP subnets in classify conditions

Parameter Configuration

The parameters appear in the configuration for:

- A policy group called tierpolicy that classifies packets based on source and destination subnets and applies a rate limit action to those packets. The tierpolicy policy group contains three local parameters:
 - inside—Parameter of type network; used to specify a subnet
 - outside—Parameter of type network; used to specify a subnet
 - qos—Parameter of type rate; used to scale the rate limiter
- A service called GoldMetered, that has tierpolicy as the policy group. The GoldMetered service includes the following parameter substitution:
 - qos—Fix to 50 % of the interface_speed parameter. (interface_speed is a global runtime parameter that the SAE fills in with the actual speed of the router interface.)
 - dept—Create a parameter called dept that is parameter type (role) network.
 - outside—Set to dept (short for department), which effectively renames the outside parameter to dept.
 - inside—Set to any.
- An enterprise subscriber that uses the following parameter substitution:
 - eng—Create a parameter called eng (short for engineering department) that is parameter type (role) network, and set the value to 192.0.2.22/28.
- A subscriber subscription to the GoldMetered service that has the following parameter substitution:
 - dept—Set to eng.

Parameter Values After Value Acquisition

After the SRC software has gone through the parameter value acquisition process, the three original parameters in the tierpolicy policy group have the following values:

- inside = 0.0.0.0/0

This value was acquired from the global parameter any that was defined in the service definition

- outside = 192.0.2.22/28

This value was acquired as follows:

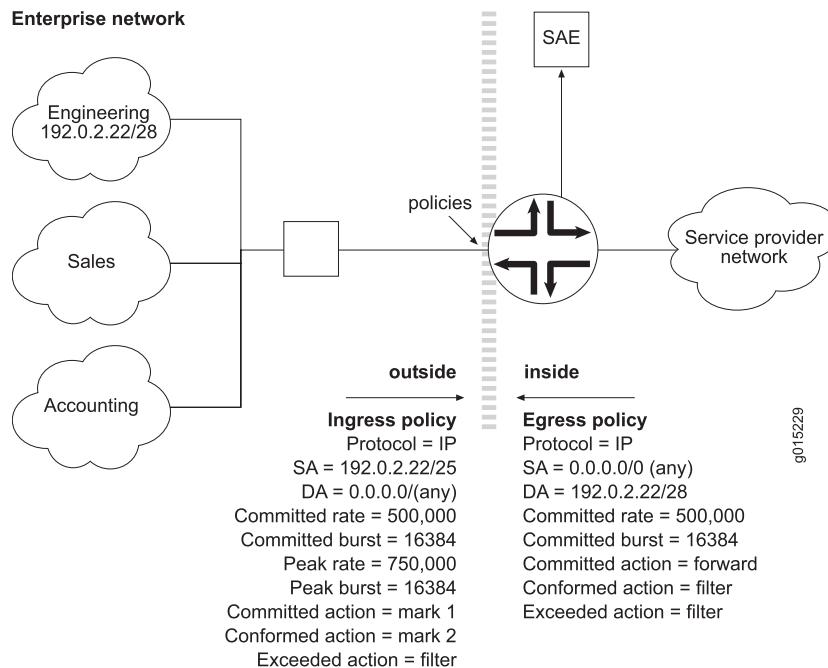
- outside = dept—Acquired from the service definition
- dept = eng—Acquired from the subscription
- eng = 192.0.2.22/28—Acquired from the enterprise subscriber definition
- qos = 500,000

This value was acquired from the service definition where the value of qos was set to 50 % of the interface_speed parameter. An interface_speed value of 1,000,000 was acquired from the router. If qos = 50 % of the interface speed, then the qos value is 500,000.

The rest of the rate-limit values are calculated based on the 500,000 value of qos.

Figure 15 on page 227 shows the values of the ingress and egress policies that are applied to the router in our sample network.

Figure 15: Policies Applied to the Sample Network



Configuration

Configure a policy, service, subscriber, and subscription to use parameter value acquisition:

- Configuring the Default Value for a Global Parameter on page 228
- Configuring a Policy Group on page 228
- Configuring a Service on page 236
- Creating an Enterprise Subscriber on page 238
- Subscribing ABCInc to the GoldMetered Service on page 239

Configuring the Default Value for a Global Parameter

Configure the global parameter any which is used in the policy configuration.

CLI Quick Configuration To quickly configure the global parameter any, copy the following commands into a text editor, and modify them as needed; then load the configuration from the file.

```
[edit]
set policies global-parameters any default-value 0.0.0.0/0
set policies global-parameters any type network
```

Step-by-Step Procedure To configure the global parameter any:

1. From configuration mode, enter the global parameter configuration for the any parameter.

```
[edit]
user@host# edit policies global-parameters any
```

2. (Optional) Configure a default value that the policy engine uses if no other values are provided during the parameter value acquisition process.

See “Parameter Types” on page 82 for valid values of each parameter type.

```
[edit policies global-parameters any]
user@host# set default-value 0.0.0.0/0
```

3. (Optional) Type of attribute for which you can use the parameter.

```
[edit policies global-parameters any]
user@host# set type network
```

Configuring a Policy Group

Configure the policy group tierpolicy to specify bandwidth for incoming and outgoing traffic.

CLI Quick Configuration To quickly configure the global parameter any, copy the following commands into a text editor, and modify them as needed; then load the configuration from the file.

[edit]

```
set policies folder ent group tierpolicy
set policies folder ent group tierpolicy local-parameters qos
set policies folder ent group tierpolicy local-parameters qos type rate
```

```
set policies folder ent group tierpolicy local-parameters outside
set policies folder ent group tierpolicy local-parameters outside type network
set policies folder ent group tierpolicy local-parameters outside default-value any
set policies folder ent group tierpolicy local-parameters inside
set policies folder ent group tierpolicy local-parameters inside type network
set policies folder ent group tierpolicy local-parameters inside default-value any
```

```
set policies folder ent group tierpolicy list egrules
set policies folder ent group tierpolicy list egrules role junose-ipv4
set policies folder ent group tierpolicy list egrules applicability output
```

```
set policies folder ent group tierpolicy list ingrules
set policies folder ent group tierpolicy list ingrules role junose-ipv4
set policies folder ent group tierpolicy list ingrules applicability input
```

```
set policies folder ent group tierpolicy list egrules rule eglimit
set policies folder ent group tierpolicy list egrules rule eglimit type junose-ipv4
set policies folder ent group tierpolicy list egrules rule eglimit precedence 1000
set policies folder ent group tierpolicy list egrules rule eglimit accounting
```

```
set policies folder ent group tierpolicy list egrules rule eglimit traffic-condition cond
set policies folder ent group tierpolicy list egrules rule eglimit traffic-condition cond
source-network group-network network-specifier inside
set policies folder ent group tierpolicy list egrules rule eglimit traffic-condition cond
destination-network group-network network-specifier outside
```

```
set policies folder ent group tierpolicy rate-limit ratelimit
set policies folder ent group tierpolicy type two-rate
```

```
set policies folder ent group tierpolicy list egrules rule eglimit rate-limit ratelimit
committed-rate qos
set policies folder ent group tierpolicy list egrules rule eglimit rate-limit ratelimit
committed-burst "max(qos*0.1, 16384)"
set policies folder ent group tierpolicy rate-limit ratelimit committed-action forward
set policies folder ent group tierpolicy rate-limit ratelimit exceed-action filter
set policies folder ent group tierpolicy rate-limit ratelimit conformed-action filter
set policies folder ent group tierpolicy rate-limit ratelimit exceed-action filter
```

```
set policies folder ent group tierpolicy list ingrules rule inglimit
set policies folder ent group tierpolicy list ingrules rule inglimit type junose-ipv4
set policies folder ent group tierpolicy list ingrules rule inglimit precedence 1000
set policies folder ent group tierpolicy list ingrules rule inglimit accounting
```

```
set policies folder ent group tierpolicy list ingrules rule inglimit traffic-condition ent
set policies folder ent group tierpolicy list ingrules rule inglimit traffic-condition ent
source-network group-network network-specifier outside
```

```
set policies folder ent group tierpolicy list ingrules rule inglimit traffic-condition ent
destination-network group-network network-specifier inside
```

```
set policies folder ent group tierpolicy list ingrules rule inglimit rate-limit rateLimit
set policies folder ent group tierpolicy list ingrules rule inglimit rate-limit rateLimit
type two-rate
```

```
set policies folder ent group tierpolicy list ingrules rule inglimit rate-limit rateLimit
committed-rate qos
```

```
set policies folder ent group tierpolicy list ingrules rule inglimit rate-limit rateLimit
committed-burst "max(qos*0.1, 16384)"
```

```
set policies folder ent group tierpolicy list ingrules rule inglimit rate-limit rateLimit
peak-rate qos*1.5
```

```
set policies folder ent group tierpolicy list ingrules rule inglimit rate-limit rateLimit
committed-action mark mark-info value 1
```

```
set policies folder ent group tierpolicy list ingrules rule inglimit rate-limit rateLimit
```

```
set policies folder ent group tierpolicy list ingrules rule inglimit rate-limit rateLimit
conformed-action mark mark-info value 2
```

```
set policies folder ent group tierpolicy list ingrules rule inglimit rate-limit rateLimit
exceed-action filter
```

```
set policies folder ent group tierpolicy list ingrules rule inglimit rate-limit rateLimit
peak-burst "max(qos*1.5*0.1, 16384)"
```

Step-by-Step Procedure To create and configure a policy group named tierpolicy:

1. Create the tiergroup policy.

```
[edit]
user@host# edit policies folder ent group tierpolicy
```

2. Create local parameters, which are parameters that will be used only with tierpolicy.

3. qos—Rate parameter

```
[edit policies folder ent group tierpolicy]
user@host# edit local-parameters qos
[edit policies folder ent group tierpolicy local-parameters qos]
user@host# set type rate
```

- outside—Network parameter with a default value of any; any is a global parameter with value 0.0.0.0/0, which matches any network

```
[edit policies folder ent group tierpolicy]
user@host# edit local-parameters outside
[edit policies folder ent group tierpolicy local-parameters outside]
user@host# set type network
[edit policies folder ent group tierpolicy local-parameters outside]
user@host# set default-value any
```

- inside—Network parameter with a default value of any; any is a global parameter with value 0.0.0.0/0, which matches any network


```
[edit policies folder ent group tierpolicy]
user@host# edit local-parameters inside
[edit policies folder ent group tierpolicy local-parameters inside]
user@host# set type network
[edit policies folder ent group tierpolicy local-parameters inside]
user@host# set default-value any
```

- Create a policy lists for egress side of the interface.

```
[edit policies folder ent group tierpolicy]
user@host# edit list egrules

[edit policies folder ent group tierpolicy list egrules]
user@host# set role junose-ipv4

[edit policies folder ent group tierpolicy list egrules]
user@host# set applicability output
```

- Create a policy list, for the ingress side of the interface.

```
[edit policies folder ent group tierpolicy]
user@host# edit list ingrules

[edit policies folder ent group tierpolicy list ingrules]
user@host# set role junose-ipv4

[edit policies folder ent group tierpolicy list ingrules]
user@host# set applicability input
```

- Create a policy rule for egress traffic.

```
[edit policies folder ent group tierpolicy list egrules]
user@host# edit rule eglimit

[edit policies folder ent group tierpolicy list egrules rule eglimit]
user@host# set type junose-ipv4

[edit policies folder ent group tierpolicy list egrules rule eglimit]
user@host# set precedence 1000

[edit policies folder ent group tierpolicy list egrules rule eglimit]
user@host# set accounting
```

- In the egress policy rule, which applies to traffic coming from the service provider network to the enterprise, create a condition that matches IP packets on source and destination networks:

- source network = inside
- destination network = outside

```
[edit policies folder ent group tierpolicy list egrules rule eglimit]
```

```
user@host# edit traffic-condition cond
```

```
[edit policies folder ent group tierpolicy list egrules rule eglimit traffic-condition
cond ]
```

```
user@host# set source-network group-network network-specifier inside
```

```
[edit policies folder ent group tierpolicy list egrules rule eglimit traffic-condition
cond ]
```

```
user@host# set destination-network group-network network-specifier outside
```

- Also in the egress policy rule, create a rate-limit action and set the type to the runtime parameter two-rate.

```
[edit policies folder ent group tierpolicy list egrules rule eglimit
user@host# edit rate-limit ratelimit
```

```
[edit policies folder ent group tierpolicy list egrules rule eglimit rate-limit ratelimit]
user@host# set type two-rate
```

- Configure the rate-limit action in the egress policy rule to do the following:
- Set the committed rate to the qos parameter.

```
[edit policies folder ent group tierpolicy list egrules rule eglimit rate-limit ratelimit]
user@host# set committed-rate qos
```

- Set the committed burst to the maximum of either 800 ms burst at committed rate in bytes ($\text{qos} \times 0.1$) or 16384.

```
[edit policies folder ent group tierpolicy list egrules rule eglimit rate-limit ratelimit]
user@host# set committed-burst "max(qos*0.1, 16384)"
```

- Use the default peak burst rate of 16384.
- Forward all committed traffic.

```
[edit policies folder ent group tierpolicy list egrules rule eglimit rate-limit ratelimit]
user@host# set committed-action forward
```

- Filter all uncommitted traffic.

```
[edit policies folder ent group tierpolicy list egrules rule eglimit rate-limit ratelimit]
user@host# set exceed-action filter
```

```
[edit policies folder ent group tierpolicy list egrules rule eglimit rate-limit ratelimit]
user@host# set conformed-action filter
```

```
[edit policies folder ent group tierpolicy list egrules rule eglimit rate-limit ratelimit]
user@host# set exceed-action filter
```

- Create a policy rule for ingress traffic.

```
[edit policies folder ent group tierpolicy list ingrules]
user@host# edit rule inglimit
```

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit]
user@host# set type junose-ipv4
```

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit]
user@host# set precedence 1000
```

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit]
user@host# set accounting
```

- In the ingress policy rule, which applies to traffic coming from the enterprise network, create a condition that matches IP packets on source and destination networks:
- source network = outside
- destination network = inside

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit]
user@host# edit traffic-condition ent
```

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit traffic-condition
ent]
user@host# set source-network group-network network-specifier outside
```

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit traffic-condition
ent]
user@host# set destination-network group-network network-specifier inside
```

- Also in the ingress policy rule, create a rate-limit action and set the type to the runtime parameter two-rate.

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit]
user@host# edit rate-limit rateLimit
```

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit rate-limit rateLimit]
user@host# set type two-rate
```

- Configure the rate-limit action in the ingress policy rule to do the following:
- Set the committed rate to the qos local parameter.

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit rate-limit rateLimit]
user@host# set committed-rate qos
```

- Set the committed burst to either 800 ms burst or at the committed rate in bytes ($\text{qos} \times 0.1$) or 16384.

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit rate-limit rateLimit]
user@host# set committed-burst "max(qos*0.1 , 16384)"
```

- Scale the peak rate and burst by 1.5.

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit rate-limit rateLimit]
user@host# set peak-rate qos*1.5
```

- Mark committed and conformed traffic with different marks (1 and 2).

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit rate-limit rateLimit]
user@host# set committed-action mark mark-info value 1
[edit policies folder ent group tierpolicy list ingrulers rule inglimit rate-limit rateLimit]
user@host# set conformed-action mark mark-info value 2
```

- Drop all traffic that exceeds the rate limit.

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit rate-limit rateLimit]
user@host# set exceed-action filter
```

- Set the peak burst rate to the maximum of either 800 ms burst of one and a half times the committed rate in bytes ($qos * 1.5$) or 16384.

```
[edit policies folder ent group tierpolicy list ingrulers rule inglimit rate-limit rateLimit]
user@host# set peak-burst "max(qos*1.5*0.1, 16384)"
```

Configuration Results

```
[edit policies folder ent group tierpolicy]
user@host# show
description "This is a service policy for services that rate limit and account
for traffic to and from the service provider's network. It is parameterized on
the subnets inside and outside the service provider's network between which the
traffic flows. It is also parameterized on a number which is used to scale ingress
and egress rate limit rules. ";
local-parameters {
  qos {
    description " Scaling factor to apply to the rate limits on the traffic between
inside and outside";
    type rate;
  }
  outside {
    description "the subnet outside the service provider's network";
    default-value any;
    type network;
  }
  inside {
    description "the subnet inside the service provider's network";
    default-value any;
    type network;
  }
}
list egrules {
  role junose-ipv4;
  applicability output;
  rule eglimit {
    type junose-ipv4;
    precedence 1000;
    accounting;
    rate-limit ratelimit {
      committed-action {
        forward {
        }
      }
    }
  }
}
```

```

    }
    conformed-action {
        filter {
        }
    }
    exceed-action {
        filter {
        }
    }
    type two_rate;
    committed-rate qos;
    committed-burst "max(qos*0.1, 16384)";
    peak-rate qos*1.5;
    peak-burst 16384;
    description "committed rate is \"qos\" parameter, burst is 800ms burst at
committed rate (*0.1 remember rates are bits per second, bursts are bytes)
drop all uncommitted traffic. Max with 16384 to make sure burst is not too small
for slow interfaces.
";
}
traffic-condition cond {
    source-network {
        group-network {
            network-specifier inside;
        }
    }
    destination-network {
        group-network {
            network-specifier outside;
        }
    }
}
description "rule to limit egress traffic";
}
}
list ingrules {
    role junose-ipv4;
    applicability input;
    rule inglimit {
        type junose-ipv4;
        precedence 1000;
        accounting;
        rate-limit rateLimit {
            committed-action {
                mark {
                    mark-info {
                        value 1;
                    }
                }
            }
            conformed-action {
                mark {
                    mark-info {
                        value 2;
                    }
                }
            }
            exceed-action {
                filter {
                }
            }
        }
    }
}

```

```

        type two_rate;
        committed-rate qos;
        committed-burst "max(qos*0.1, 16384)";
        peak-rate qos*1.5;
        peak-burst "max(qos*1.5*0.1, 16384)";
        description "committed rate is \"qos\" parameter, burst is 800ms burst at
        committed rate (*0.1 remember rates are bits per second, bursts are bytes). Max
        with 16384 to make sure burst is not too small for slow interfaces.peak rate and
        burst are scaled by 1.5. mark committed and conformed traffic with different
        marks, drop all excess traffic
        ";
    }
    traffic-condition ent {
        source-network {
            group-network {
                network-specifier outside;
            }
        }
        destination-network {
            group-network {
                network-specifier inside;
            }
        }
    }
    description "rule to limit ingress traffic";
}

```

Configuring a Service

Configure a service that provides a gold-level quality of service to subscribers.

CLI Quick Configuration To quickly configure a service copy the following commands into a text editor, and modify them as needed; then load the configuration from the file.

```

[edit]
set services
set services scope EntJunose
set services scope EntJunose service GoldMetered
set services scope EntJunose service GoldMetered type normal
set services scope EntJunose service GoldMetered category "Quality of Service"
set services scope EntJunose service GoldMetered policy-group /ent/tierpolicy
set services scope EntJunose service GoldMetered radius-class GoldMetered
set services scope EntJunose service GoldMetered parameter substitution
[ "dept:network//the subnet of the department to apply the service to"
"!inside:network = any//always apply to any subnet inside the service provider"
"!outside:network = dept//rename outside policy parameter to dept" "!qos =
interface_speed*0.5//gold qos is 50% of interface speed" ]

```

Step-by-Step Procedure To configure a service that uses the policy tierpolicy:

1. Create a service called GoldMetered, and assign tierpolicy as the policy group.

```

[edit]
user@host# edit services

```

```

[edit services]

```

```
user@host# edit scope EntJunose
```

```
[edit services scope EntJunose]
user@host# edit service GoldMetered
```

```
[edit services scope EntJunose service GoldMetered]
user@host# set type normal
```

```
[edit services scope EntJunose service GoldMetered]
user@host# set category "Quality of Service"
```

```
[edit services scope EntJunose service GoldMetered]
user@host# set policy-group /ent/tierpolicy
```

```
[edit services scope EntJunose service GoldMetered]
user@host# set radius-class GoldMetered
```

2. Edit the parameter for the GoldMetered service, and add the following substitutions:
 - dept—Create a parameter called dept that is parameter type (role) network. This is the subnet of the department that the service will apply to.
 - qos—Fix the qos parameter to 50 % of the interface_speed parameter. (interface_speed is a global runtime parameter that the SAE fills in with the actual speed of the router interface).
 - outside—Set the outside parameter to the value dept, which effectively renames the outside parameter to dept.
 - inside—Set the inside parameter to a value of any, which applies to any subnet inside the service provider's network.

```
[edit services scope EntJunose service GoldMetered]
user@host# set parameter substitution [ "dept:network//the subnet of the
department to apply the service to" "!inside:network = any//always apply
to any subnet inside the service provider" "!outside:network = dept//rename
outside policy parameter to dept" "!qos = interface_speed*0.5//gold qos is
50% of interface speed" ]
```

Configuration Results

```
[edit services scope EntJunose service GoldMetered]
user@host# show
description "Provides gold level quality of service to given enterprise department
  subnet charged on volume";
type normal;
category "Quality of Service";
policy-group /ent/tierpolicy;
radius-class GoldMetered;
status active;
parameter {
  substitution [ "dept:network//the subnet of the department to apply the servic
e to" "!inside:network = any//always apply to any subnet inside the service prov
```

```

ider" "!outside:network = dept//rename outside policy parameter to dept" "!qos =
  interface_speed*0.5//gold qos is 50% of interface speed" ];
}

```

Creating an Enterprise Subscriber

Create the eng parameter for use in parameter substitution. this parameter represents an enterprise subscriber. You can configure the substitution in the SRC CLI, the sample enterprise service portal, or the C-Web interface.

CLI Quick Configuration

To quickly configure the global parameter any, copy the following commands into a text editor, and modify them as needed; then load the configuration from the file.

```

[edit]
set subscribers retailer default subscriber-folder local enterprise ABCInc substitution
[ " acct : network = 208.93.36.80 / 28" "eng : network = 208.93.36.6 4 / 28"
]
set subscribers retailer default subscriber-folder local enterprise ABCInc substitution
[ "acct : network = 208.93.36.80 / 28" "eng : network = 208.93.36.64 / 28"
]

```

Step-by-Step Procedure

To create a parameter called eng in an existing enterprise:

1. Create the eng parameter with parameter type (role) network, and set the value of eng to 192.0.2.22/28.

```

[edit subscribers retailer default subscriber-folder local enterprise ABCInc]
user@host# set substitution [ "acct : network = 208.93.36.80 / 28" "eng :
network = 208.93.36.6 4 / 28" ]

```

2. Create the eng parameter as part of the subscriber definition.

- To create the eng parameter with the SRC CLI:

```

[edit subscribers retailer default subscriber-folder local enterprise ABCInc]
user@host# set substitution [ "acct : network = 208.93.36.80 / 28" "eng :
network = 208.93.36.64 / 28" ]

```


- To create the eng parameter in the sample enterprise service portal, select the **Departments** tab, add eng to the department field, and enter 192.0.2.22/28 as the network address of the department.



Virneo Enterprise Portal Log out

Navigation

- ent-admin
 - default
 - local
 - ABCInc
 - Boca
 - Backup
 - Primary
 - Boston
 - Montreal
 - Ottawa
 - PrimaryAccess
 - Acme
 - retailer-one
 - retailer-two
 - SP
 - virtual-SP

Refresh

default ▸ local ▸ ABCInc ▸ Boca ▸ **Primary** ▸

Services	Subscriptions	Sessions	Departments	Managers
Department	Department network	Locked		
eng	192.0.2.22/28 <small>(From enterprise ABCInc)</small>	<input type="checkbox"/>	Apply Delete Reset	
acct	208.93.36.80/28 <small>(From enterprise ABCInc)</small>	<input type="checkbox"/>	Apply Delete Reset	
<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	Create	

© Juniper Networks 2003-2004

Configuration Results

```
[edit subscribers retailer default subscriber-folder local enterprise ABCInc]
user@host# show
...
substitution [ "acct : network = 208.93.36.80 / 28" "eng : network = 208.93.36.6
4 / 28" ];
...
subscription GoldSecured {
    status active;
    activation manual;
    substitution "!dept : network = eng";
}
```

Subscribing ABCInc to the GoldMetered Service

Subscribe to the GoldMetered service.

Step-by-Step Procedure

To subscribe the ABCInc subscriber to the GoldMetered service through the sample enterprise service portal.

1. Select **ABCInc.** in the navigation pane.
2. Select the **Services** tab.

The Services pane appears.



Virneo Enterprise Portal

[Log out](#)

Navigation

ent-admin

default

local

ABCInc

Boca

Boston

Montreal

Ottawa

PrimaryAccess

Acme

retailer-one

retailer-two

SP

virtual-SP

Refresh

default

local

ABCInc

Services

Subscriptions

Departments

Managers

Service	Current local subscriptions	New local subscription name	
Internet-Gold		<input type="text"/>	Subscribe
News		<input type="text"/>	Subscribe
Video-Bronze		<input type="text"/>	Subscribe
Audio-Bronze		<input type="text"/>	Subscribe
PingDoSProtect	[unnamed]	<input type="text"/>	Subscribe
GoldMetered	[unnamed]	<input type="text"/>	Subscribe
GoldSecured		<input type="text"/>	Subscribe
BronzeMetered	[unnamed]	<input type="text"/>	Subscribe
Internet-Silver		<input type="text"/>	Subscribe

3. Click **Subscribe** in the GoldMetered service row.
4. Select the **Subscriptions** tab.

The Subscriptions pane appears.



Virneo Enterprise Portal Log out

Navigation

- ent-admin
 - default
 - local
 - ABCInc
 - Boca
 - Boston
 - Montreal
 - Ottawa
 - PrimaryAccess
 - Acme
 - retailer-one
 - retailer-two
 - SP
 - virtual-SP

Refresh

default ▶ local ▶ ABCInc ▶

Service	Subscription	Subscription details								
BronzeMetered	[unnamed]	<table border="1"> <thead> <tr> <th>Subscription Status</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>Administratively inactive.</td> <td> <input type="button" value="Activate"/> <input type="button" value="Deactivate"/> </td> </tr> <tr> <td>Not suspended.</td> <td> <input type="button" value="Unsuspend"/> <input type="button" value="Suspend"/> </td> </tr> <tr> <td>Usage</td> <td> <input type="button" value="Reporting"/> </td> </tr> </tbody> </table>	Subscription Status	Action	Administratively inactive.	<input type="button" value="Activate"/> <input type="button" value="Deactivate"/>	Not suspended.	<input type="button" value="Unsuspend"/> <input type="button" value="Suspend"/>	Usage	<input type="button" value="Reporting"/>
Subscription Status	Action									
Administratively inactive.	<input type="button" value="Activate"/> <input type="button" value="Deactivate"/>									
Not suspended.	<input type="button" value="Unsuspend"/> <input type="button" value="Suspend"/>									
Usage	<input type="button" value="Reporting"/>									
GoldMetered	[unnamed]									
PingDoSPProtect	[unnamed]									

Service Parameters (use checkbox to lock value)	Action
dept = <input type="text" value="any"/> <input checked="" type="checkbox"/>	<input type="button" value="Apply"/> <input type="button" value="Delete"/> <input type="button" value="Reset"/>

any
eng
acct

5. In the dept = field of the Service Parameters box, set the value of the dept parameter to eng.

- Related Topics**
- Parameters and Substitutions on page 209
 - Value Acquisition for Single Subscriptions on page 210
 - Formatting Substitutions on page 214
 - Adding Comments to Substitutions on page 223
 - Validating Substitutions on page 224
 - Adding a Normal Service (SRC CLI) on page 5

Part 4

Index

- Index on page 245

Index

A

access policy, examples.....	195
DHCP	
SRC CLI.....	195
PPP	
SRC CLI.....	197
action threshold, service schedules	
overview.....	35
setting	
SRC CLI.....	43
actions. <i>See</i> policy actions	
aggregate services.....	3
adding	
SRC CLI.....	15
before you configure	
SRC CLI.....	14
fragment services.....	10
infrastructure services.....	20
mandatory services.....	11
Python expressions.....	18
redundancy.....	11
sessions.....	11
activation.....	12
deactivation.....	12
monitoring.....	12
timers, configuring	
SRC CLI.....	17
apply-groups statement, JUNOS routing	
platforms.....	106

C

captive portal	
using with next-hop action	
SRC CLI.....	166
classify-traffic condition.....	66
application protocol	
defining, SRC CLI.....	144
map expressions, SRC CLI.....	147
configuring	
SRC CLI.....	113
destination grouped network, configuring	
SRC CLI.....	122

destination network, configuring	
SRC CLI.....	121
expanded classifiers.....	67
configuring, SRC CLI.....	115
extended classifiers.....	76
configuring, SRC CLI.....	116
ICMP conditions, setting	
SRC CLI.....	134
IGMP conditions, setting	
SRC CLI.....	135
IPSec conditions, setting	
SRC CLI.....	136
JUNOS filter conditions, setting	
SRC CLI.....	139
JUNOS secondary input policy conditions, setting	
SRC CLI.....	142
match direction, setting	
SRC CLI.....	118
multiple classifiers.....	67
packet length, setting	
SRC CLI.....	123
PCMM I02 and I03.....	76
configuring, SRC CLI.....	116
port definitions, overview	
SRC CLI.....	117
protocol conditions with parameters, setting	
SRC CLI.....	127
protocol conditions with ports, setting	
SRC CLI.....	124
protocol conditions, setting	
SRC CLI.....	123
route class, configuring	
SRC CLI.....	142
source grouped network, configuring	
SRC CLI.....	120
source network, setting	
SRC CLI.....	119
TCP conditions, setting	
SRC CLI.....	131
ToS byte conditions, setting	
SRC CLI.....	136
color actions.....	68
configuring	
SRC CLI.....	149
color mark actions.....	68, 150
controlled load service, FlowSpec.....	78

conventions	
notice icons.....	xxi
text.....	xxi
CoS (class of service)	
ToS byte, setting	
SRC CLI.....	138
customer support.....	xxv
contacting JTAC.....	xxv

D

Data-over-Cable Service Interface Specifications. <i>See</i>	
DOCSIS	
default policies	
example	
SRC CLI.....	195
DHCP (Dynamic Host Configuration Protocol)	
access policy example	
SRC CLI.....	195
Differentiated Services code point, ToS byte	
SRC CLI.....	138
DOCSIS policy actions.....	69
configuring	
SRC CLI.....	151
documentation set	
comments on.....	xxv
drop profile maps	
configuring	
SRC CLI.....	184
drop probability, setting	
SRC CLI.....	184
fill level, setting	
SRC CLI.....	184
DSCP (Differentiated Services code point), ToS byte	
SRC CLI.....	138

E

effective period, service schedules.....	38
example-simple.....	30, 51, 53, 55, 195, 199, 203, 224
exclusions to service schedule.....	39
defining	
SRC CLI.....	47
expanded classifiers.....	67
configuring	
SRC CLI.....	115
expressions	
map, application protocol conditions	
SRC CLI.....	147
parameter definitions.....	215
extended classifiers, PCMM.....	76
configuring	
SRC CLI.....	116

F

filter actions.....	69
configuring	
SRC CLI.....	156
FlowSpec actions.....	69
configuring	
SRC CLI.....	157
forward actions.....	69
configuring	
SRC CLI.....	159
forwarding class actions.....	69
configuring	
SRC CLI.....	160
fragment services.....	10
configuring	
SRC CLI.....	15

G

gates, PCMM.....	76
gateSpec actions.....	69
configuring	
SRC CLI.....	161
global parameters.....	81
configuring	
SRC CLI.....	101
predefined.....	90
viewing with SRC CLI.....	101
runtime.....	90
types.....	82
guaranteed service, FlowSpec.....	78

I

infrastructure services.....	3, 20
------------------------------	-------

J

JUNOS ASP policy rules.....	64
NAT actions.....	69
configuring, SRC CLI.....	165
network, specifying.....	122
SRC CLI.....	120, 122
stateful firewall actions, configuring	
SRC CLI.....	186
JUNOS filter policy rules.....	64
conditions, setting	
SRC CLI.....	139
JUNOS policer policy rules.....	64
policer actions.....	69
configuring, SRC CLI.....	169
JUNOS port mirror policy rules	
traffic mirror actions.....	70
JUNOS routing platforms	
policy features	
rate-shaping.....	64

JUNOS scheduler policy rules.....	64, 184
actions.....	70
configuring, SRC CLI.....	182
QoS conditions, configuring	
SRC CLI.....	147
<i>See also</i> drop profile maps	
JUNOS shaping policy rules.....	64
JUNOSe IPv6 policy rules	
network, specifying	
SRC CLI.....	120, 122
JUNOSe secondary input policy rules	
conditions, setting	
SRC CLI.....	142

L

local parameters.....	81
configuring	
SRC CLI.....	103
types.....	82
loss priority actions.....	69
configuring	
SRC CLI.....	163

M

manuals	
comments on.....	xxv
map expressions	
application protocol conditions	
SRC CLI.....	147
substitutions.....	218
mark actions.....	69
configuring	
SRC CLI.....	164
multi-tsak.....	32
multiple classifiers, policies.....	67
mutex group.....	26
adding	
SRC CLI.....	27

N

NAT (Network Address Translation) policies	
actions.....	69
configuring, SRC CLI.....	165
application protocol condition	
defining, SRC CLI.....	144
map expressions, SRC CLI.....	147
next-hop actions.....	69
captive portal feature	
SRC CLI.....	166
configuring	
SRC CLI.....	166

next-interface actions.....	69
configuring	
SRC CLI.....	167
next-rule actions.....	69
configuring	
SRC CLI.....	168
non-real-time polling service.....	74
notice icons.....	xxi
NRTPS (non-real-time polling service).....	74

O

operators in substitution expressions.....	218
--	-----

P

packet loss priority. <i>See</i> loss priority actions	
PacketCable Multimedia Specifications. <i>See</i> PCMM	
parameter names	
substitutions.....	215
parameter value acquisition.....	209
example.....	224
multiple subscriptions.....	211
single subscriptions.....	210
<i>See also</i> substitutions	
parameter values, setting in services.....	8
parameters.....	215
defining.....	212
definition.....	209
fixing.....	209
global. <i>See</i> global parameters	
local. <i>See</i> local parameters	
ranking sources.....	209
runtime. <i>See</i> runtime parameters	
types.....	82
<i>See also</i> substitutions	
parent groups.....	64, 71, 140, 175
PCMM policies	
classifiers.....	76
client type 1 support.....	75
conditions and actions supported.....	66
DOCSIS parameters.....	77
configuring, SRC CLI.....	151
extended classifiers.....	76
configuring, SRC CLI.....	116
FlowSpec parameters	
configuring, SRC CLI.....	157
controlled load service.....	78
guaranteed service.....	78
request specification (RSpec).....	78
traffic specification (TSpec).....	78
gate.....	76
gateSpec parameters, configuring	
SRC CLI.....	161
102 and 103 classifiers.....	76
marking packets.....	79

proxied QoS with policy push.....	75	rate limit hierarchy	
service class name		configuring, SRC CLI.....	175
configuring, SRC CLI.....	185	overview.....	71
service flow scheduling types.....	73	parent-group reference, SRC CLI.....	140
SessionClassId.....	76	rate limit types	
traffic profiles.....	77	configuring, SRC CLI.....	178
permanent service.....	4	reject.....	70
configuring		configuring, SRC CLI.....	180
SRC CLI.....	7	routing instance.....	70
plug-ins		configuring, SRC CLI.....	181
authorization.....	44	scheduler.....	70
policer actions.....	69	configuring, SRC CLI.....	182
configuring		service class name.....	70
SRC CLI.....	169	configuring, SRC CLI.....	185
policies		stateful firewall.....	70
defining parameters in repository.....	209	configuring, SRC CLI.....	186
policing policies		template activation	
example		configuring, SRC CLI.....	188
SRC CLI.....	201	traffic class.....	70
policy actions.....	62	configuring, SRC CLI.....	189
color.....	68	traffic mirror.....	70
configuring, SRC CLI.....	149	configuring, SRC CLI.....	190
color mark.....	68, 150	traffic-shape.....	70
combining.....	70	configuring, SRC CLI.....	191
configuring.....	148	types.....	68
DOCSIS.....	69	user packet class.....	70
configuring, SRC CLI.....	151	configuring, SRC CLI.....	192
filter.....	69	policy components.....	60
configuring, SRC CLI.....	156	policy decision point, description.....	61
FlowSpec.....	69	Policy Editor.....	61
configuring, SRC CLI.....	157	policy enforcement point, description.....	62
forward.....	69	policy engine.....	61
configuring, SRC CLI.....	159	policy repository.....	62
forwarding class.....	69	policy conditions.....	62, 66
configuring, SRC CLI.....	160	policy rules supported.....	65
gateSpec.....	69	types.....	66
configuring, SRC CLI.....	161	<i>See also</i> classify-traffic condition\	
loss priority.....	69	policy engine.....	61
configuring, SRC CLI.....	163	policy examples	
mark.....	69	access policy	
configuring, SRC CLI.....	164	SRC CLI.....	195
NAT.....	69	premium service	
configuring, SRC CLI.....	165	SRC CLI.....	203
next hop.....	69	tiered Internet service	
configuring, SRC CLI.....	166	SRC CLI.....	199
next interface.....	69	policy folders.....	63
configuring, SRC CLI.....	167	configuring	
next rule.....	69	SRC CLI.....	107
configuring, SRC CLI.....	168	policy groups.....	63
policer.....	69	configuring	
configuring, SRC CLI.....	169	SRC CLI.....	108
policy rules supported.....	65	policy lists.....	63
QoS profile attachment.....	69	configuring	
configuring, SRC CLI.....	172	SRC CLI.....	108
rate limit.....	70		
configuring, SRC CLI.....	173		

policy management	
bandwidth management.....	59
overview.....	59
packet logging.....	60
packet mirroring.....	59
packet tagging.....	59
policy routing.....	59
QoS classification and marking.....	59
RADIUS support.....	59
security.....	59
policy objects	
organization.....	63
policy overview	
actions. <i>See</i> policy actions	
conditions. <i>See</i> classify-traffic condition\	
policy object organization.....	63
policy repository, description.....	62
policy rules.....	64
actions supported.....	65
conditions supported.....	65
configuring	
SRC CLI.....	110
JUNOS Adaptive Services PIC (ASP). <i>See</i> JUNOS	
ASP policy rules	
JUNOS filter. <i>See</i> JUNOS filter policy rules	
JUNOS policer. <i>See</i> JUNOS policer policy rules	
JUNOS scheduler. <i>See</i> JUNOS scheduler policy	
rules	
JUNOS shaping. <i>See</i> JUNOS shaping policy rules	
precedence	
SRC CLI.....	111
types.....	64
PPP	
access policy example	
SRC CLI.....	197
precedence	
policy rules	
SRC CLI.....	111
premium service, example	
SRC CLI.....	203
preparation time, service schedules	
overview.....	35
setting	
SRC CLI.....	43
proxied QoS with policy push.....	75

Q

QoS (quality of service)	
condition.....	66
configuring, SRC CLI.....	147
PCMM cable networks. <i>See</i> PCMM policies	
QoS parameters, configuring	
SRC CLI.....	173

QoS profile attachment actions.....	69
configuring, SRC CLI.....	172
QoS profile, configuring	
SRC CLI.....	172
QoS condition	62, 66

R

rate-limit actions.....	70
configuring	
SRC CLI.....	173
example	
SRC CLI.....	199
rate-limit hierarchy actions	
configuring	
SRC CLI.....	175
overview.....	71
rate-limit type actions	
configuring	
SRC CLI.....	178
rate-limiting, with multiple classifiers.....	67
real-time polling service. <i>See</i> RTPS	
reject actions.....	70
configuring	
SRC CLI.....	180
routing instance actions.....	70
configuring	
SRC CLI.....	181
RTPS (real-time polling service).....	74
configuring.....	152
SRC CLI.....	152
runtime parameters	
viewing with SRC CLI.....	104

S

scheduleAuth plug-in.....	44
scheduler actions.....	70, 184
configuring	
SRC CLI.....	182
<i>See also</i> drop profile maps	
scopes. <i>See</i> service scopes	
script services.....	21
adding	
SRC CLI.....	24
example	
ScriptService SPI in Java.....	24
ScriptService SPI in Jython.....	23
ScriptService interface.....	22
service class name actions.....	70
configuring	
SRC CLI.....	185
service flow scheduling types.....	73

service schedules	
action threshold, setting	
SRC CLI.....	43
authorization schedules, configuring	
SRC CLI.....	44
configuring	
SRC CLI.....	45
examples	
SRC CLI.....	51, 53, 55
exclusions, defining	
SRC CLI.....	47
guidelines.....	40
overview.....	35
action threshold.....	35
authorization schedules.....	37
configuring.....	39
effective period.....	38
event-based schedules.....	35
exclusions.....	39
one-time events.....	38
preparation time.....	36
recurring events.....	38
state-based schedules.....	37
planning.....	41
preparation time, setting	
SRC CLI.....	43
service scopes.....	28
adding	
SRC CLI.....	32
assigning services	
SRC CLI.....	32
assigning subscribers	
SRC CLI.....	29
assigning VRs	
SRC CLI.....	29
configuring	
SRC CLI.....	32
example	
SRC CLI.....	30
multiple scopes, defining	
SCR CLI.....	29
service-mgm-schedules-nonwork.....	53
services	
activate-only.....	33
adding aggregate	
SRC CLI.....	15
adding infrastructure	
SRC CLI.....	20
adding normal	
SRC CLI.....	5
adding script services	
SRC CLI.....	24
aggregate. <i>See</i> aggregate services	
assigning to service scopes	
SRC CLI.....	32
automatic activation.....	4
infrastructure. <i>See</i> infrastructure services	
mutually exclusive.....	26
overview	
C-series Controller.....	3
premium service example	
SRC CLI.....	203
restricting availability	28
restricting simultaneous activation.....	26
script. <i>See</i> script services	
setting parameter values.....	8
tiered Internet example	
SRC CLI.....	199
SessionClassId, PCMM policies.....	76
shaping rate. <i>See</i> traffic shaping	
stateful firewall policies	
actions.....	70
configuring, SRC CLI.....	186
application protocol conditions	
defining, SRC CLI.....	144
map expressions, SRC CLI.....	147
substitutions.....	214
aggregate services, configuring.....	18
comments	215
adding.....	223
definition.....	209
exceptions, raising.....	216
expressions.....	215, 223
IPv4 addresses.....	217
keywords.....	218
lists, formatting.....	217
maps, formatting.....	218
numbers, formatting.....	217
operators.....	218
parameter names, specifying.....	216
ranges.....	217
separators.....	218
strings, formatting.....	217
subordinate expressions.....	216
syntax.....	216
formatting.....	214
map expressions.....	218
mathematical expressions.....	215
parameter names.....	215
validation.....	224
<i>See also</i> parameters	
support, technical <i>See</i> technical support	
T	
technical support	
contacting JTAC.....	xxv
template activation actions	
configuring	
SRC CLI.....	188
text conventions defined.....	xxi

tiered Internet service, example	
SRC CLI.....	199
traffic mirror actions.....	70
configuring	
SRC CLI.....	190
traffic profiles, PCMM policies.....	77
traffic shape actions	
configuring	
SRC CLI.....	191
traffic shaping	
actions.....	70
policy rules.....	64
traffic-class actions.....	70
configuring	
SRC CLI.....	189
traffic-shape actions.....	70

U

UGS (unsolicited grant service).....	74
configuring	
SRC CLI.....	152
UGS-AD (unsolicited grant service with activity	
detection).....	74
configuring	
SRC CLI.....	152
unsolicited grant service. <i>See</i> UGS	
unsolicited grant with activity detection. <i>See</i> UGS-AD	
user packet class actions.....	70
configuring	
SRC CLI.....	192

V

validating	
substitutions.....	224
value acquisition for parameters	
multiple subscriptions.....	211
single subscriptions.....	210

