SRC-PE Software

Services and Policies Guide

Release 2.0.x

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

Part Number: 530-022126-01, Revision 01

SRC-PE Software Services and Policies Guide, Release 2.0.x
Writing: Justine Kangas, Helen Shaw, Linda Creed, Betty Lew
Editing: Fran Mues
Illustration: Nathaniel Woodward
Cover Design: Edmonds Design

Revision History
11 October 2007—Revision 1

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

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 Juniper Networks, Inc. and its subsidiaries (collectively “Juniper”), and 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, and updates and releases of such software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller. “Embedded Software” means Software which Juniper has embedded in the Juniper 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 the 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 software on multiple computers requires multiple licenses, regardless of whether such computers 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 the Embedded Software on non-Juniper equipment; (j) use the 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 for the Software are exclusive of taxes, withholdings, duties, or levies (collectively "Taxes"). Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software.

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

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

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

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

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

Table of Contents

Part 2 Defining Policies to Manage Traffic

About This Guide

This preface provides the following guidelines for using *SRC-PE Software Services and Policies Guide*.

Objectives

This guide describes how to manage services and policies for your Session and Resource Control (SRC) configuration. It also provides information about how the SAE acquires values for policies when it is managing JUNOSe routers or JUNOS routing platforms.

NOTE: If the information in the latest *SRC Release Notes* differs from the information in this guide, follow the *SRC Release Notes*.

Audience

This guide is intended for experienced system and network specialists working with JUNOSe 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. For users who deploy the SRC software on a Solaris platform, we also assume that readers are familiar with the Lightweight Directory Access Protocol (LDAP) and the UNIX operating system.

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

The sample screens used throughout this guide are representations of the screens that are displayed when you install and configure the SRC software. The actual screens may differ.

For convenience and clarity, the installation and configuration examples show default file paths. If you do not accept the installation defaults, your paths will vary from the examples.

Table 1 defines notice icons used in this guide. Table 2 defines text conventions used throughout the 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.

Table 2: Text Conventions

Convention	Description	Examples
Bold typeface	■ Represents keywords, scripts, options, and tools in text. ■ Represents a GUI element that the user selects, clicks, checks, or clears.	■ Specify the keyword **exp-msg**. ■ Run the **install.sh** script with the **-unconfig** option. ■ Use the **pkgadd** tool. ■ To cancel the configuration, click **Cancel**.
Bold sans serif typeface	Represents text that the user must type.	user@host# **set cache-entry-age** *cache-entry-age*

`Monospace sans serif typeface`	Represents information as displayed on your terminal’s screen, such as CLI commands in output displays.	`nic-locators {`
		`  login {`
		`    resolution {`
		`      resolver-name /realms/login/A1;`
		`      key-type LoginName;`
		`      value-type SaeId;`
		`    }`

Table 2: Text Conventions (continued)

Convention	Description	Examples
Regular sans serif typeface	■ Represents configuration statements. ■ Indicates SDX CLI commands in text. ■ Represents examples in procedures. ■ Represents URLs.	■ system ldap server { stand-alone; ■ Use the show configuration command. ■ user@host# . . . ■ http://www.juniper.net/techpubs/software/management/sdx/api-index.html
Italic sans serif typeface	Represents variables in SDX CLI commands.	user@host# **set local-address** *local-address*
Angle brackets	In text descriptions, indicate optional keywords or variables.	Another runtime variable is <gfwif>.
Key name	Indicates the name of a key on the keyboard.	Press Enter.
Key names linked with a plus sign (+) .	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
Italic typeface	■ Emphasizes words. ■ Identifies chapter, appendix, and book names. ■ Identifies distinguished names. ■ Identifies files, directories, and paths in text but not in command examples.	■ There are two levels of access: *user* and *privileged.* ■ *Chapter 2, Services.* ■ *o = Users, o = UMC* ■ The */etc/default.properties* file.
Backslash	At the end of a line, indicates that the text wraps to the next line.	Plugin.radiusAcct-1.class = \\ net.juniper.smgt.sae.plugin\\ RadiusTrackingPluginEvent
Words separated by the | symbol	Represent a choice to select one keyword or variable to the left or right of this symbol. (The keyword or variable may be either optional or required.)	diagnostic | line

Related Juniper Networks Documentation

With each SRC software release, we provide the *SRC Documentation CD,* which contains the documentation described in Table 3.

With each SRC Application Library release, we provide the *SRC Application Library CD*. This CD contains both the software applications and the *SRC Application Library Guide.*

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

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

Document	Description
Core Documentation Set	
C2000 and C4000 Hardware Guide	Describes the hardware platforms and how to install, maintain, replace, and troubleshoot them. The guide also includes specifications.
C2000 and C4000 Quick Start Guide	Describes how to get the C-series Controller up and running quickly. Intended for experienced installers who want to expedite the installation process.
SRC-PE Getting Started Guide	Describes the SRC software, how to set up an initial software configuration, 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.
SRC-PE CLI User Guide	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.
SRC-PE Network Guide: SAE, Juniper Networks Routers, NIC, and SRC-ACP	Describes how to use and configure the SAE, the NIC, and the SRC-ACP (Admission Control Plug-In) application. This guide also provides detailed information for using JUNOSe routers and JUNOS routing platforms in the SRC network.
SRC-PE Integration Guide: Network Devices, Directories, and RADIUS Servers	Describes how to integrate external components—network devices, directories, and RADIUS servers—into the SRC network. The guide provides detailed information about integrating specific models of the external components.
SRC-PE Services and Policies Guide	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.
SRC-PE Subscribers and Subscriptions Guide	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.
SRC-PE Monitoring and Troubleshooting Guide	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.
SRC-PE Solutions Guide	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.
SRC-PE CLI Command Reference, Volume 1 *SRC-PE CLI Command Reference, Volume 2*	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.
SRC-PE NETCONF API Guide	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.
SRC-PE XML API Configuration Reference	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).
SRC-PE XML API Operational Reference	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).

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

Document	Description
SRC-PE Comprehensive Index	Provides a complete index of the SRC guides, excluding the *C-series Hardware Guide*, the *SRC CLI Command Reference,* the *SRC-PE NETCONF API Guide,* the *SRC-PE XML API Configuration Reference,* and the *SRC-PE XML API Operational Reference.*
Application Library	
SRC Application Library Guide	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-SG (SOAP Gateway) 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	
SRC-PE Release Notes *SRC Application Library Release Notes*	In the *Release Notes*, 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 *Release Notes* differs from the information found in the documentation set, follow the *Release Notes.* 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 documentation, see the products documentation page on the Juniper Networks Web site at

http://www.juniper.net/

To order printed copies of this manual and other Juniper Networks technical documents, or to order a documentation CD, which contains this manual, contact your sales representative.

Documentation Feedback

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

http://www.juniper.net/techpubs/docbug/docbugreport.html

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

Requesting Support

For technical support, open a support case using the Case Manager link at

http://www.juniper.net/support/

or call 1-888-314-JTAC (from the United States, Canada, or Mexico) or 1-408-745-9500 (from elsewhere).

Part 1

Managing Services and Service Schedules

Chapter 1

Managing Services with the SRC CLI

This chapter describes how to manage services with the SRC CLI.

You can also use the following to manage services:

- To use the C-Web interface, see *SRC-PE C-Web Interface Configuration Guide, Chapter 23, Configuring Services with the C-Web Interface*
- To to configure the SRC software on a Solaris platform. See *Chapter 2, Managing Services on a Solaris Platform*.

Topics in this chapter include:

Overview of Services

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.

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
```

Before You Configure 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 *Defining Policies to Manage Traffic* on page 139.

Adding a Normal Service with the 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;
    }
    ```

Setting Parameter Values for Services with the 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 *Chapter 14, Defining and Acquiring Values for Parameters*.

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.

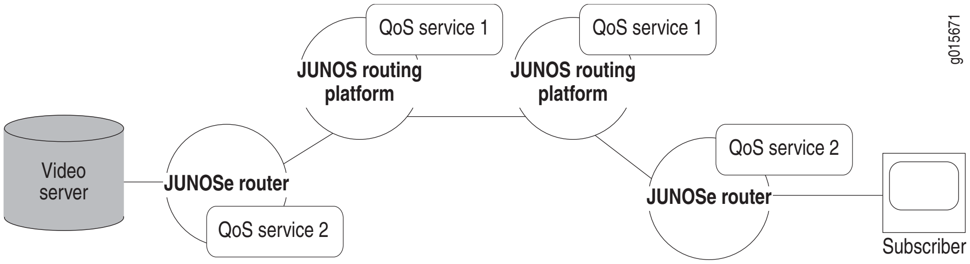
   ```
   [edit services global service Video-Gold parameter]
   user@host# show
   service-ip-address 10.10.40.16;
   service-ip-mask 255.255.255.240;
   substitution "bw = 5000000";
   ```

Aggregating Services

An aggregate service comprises a number of individual 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 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 *SRC-PE Network Guide: SAE, Juniper Networks Routers, NIC, and SRC-ACP, Chapter 19, 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 with the SRC CLI* on page 18.

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.

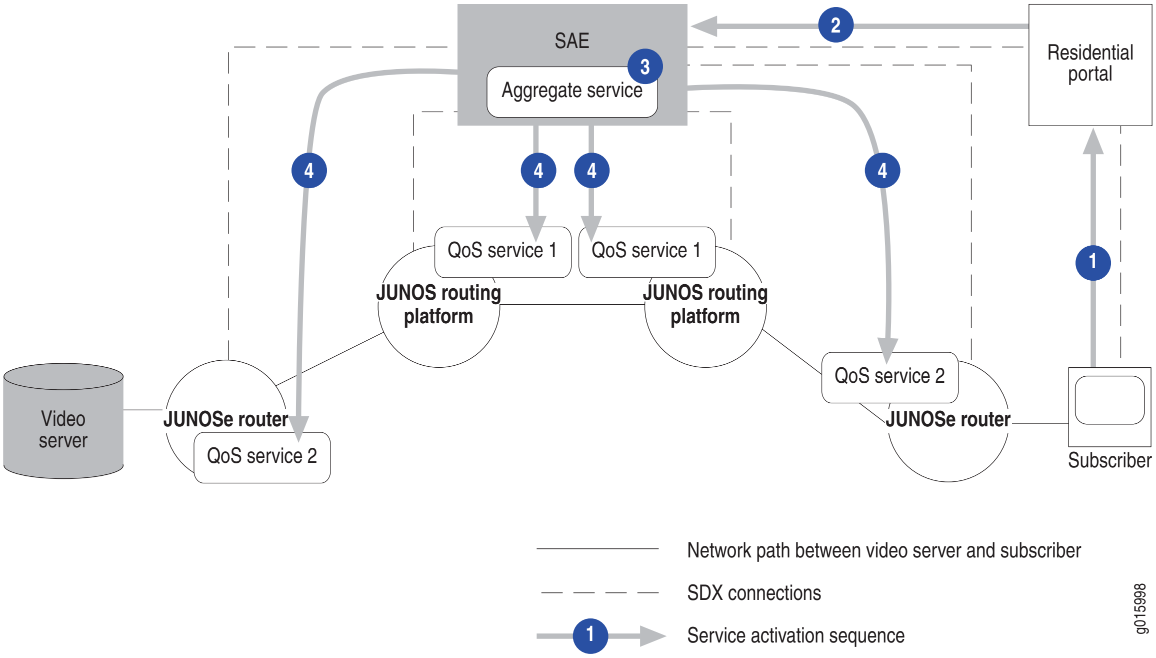
Session Monitoring

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 JUNOSe 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 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.

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.

For detailed information about service activation, see *SRC-PE Subscribers and Subscriptions Guide, Chapter 3, Subscriber Logins and Service Activation*.

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 with the 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 *Subscriber Reference Expressions for Fragment Services* on page 10.

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 *SRC-PE Getting Started Guide, Chapter 21, Setting Up an SAE with the SRC CLI*.

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.

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.

Configuring Service Fragments for an Aggregate Service with the 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;
```

Using Python Expressions in a Subscriber Reference Expression

You can compose Python expressions from one or more of the fields in Table 4 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.
loginType	The type of subscriber session, one of the following: ■ 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	Alternate name for the interface. This is the name used by the Simple Network Management Protocol (SNMP). On a JUNOSe router the format of the description is: ip<slot>/<port>.<subinterface> On a JUNOS routing platform, ifDesc is the same as interfaceName.

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

Field	Description
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
nasIp	Network access server IP address of the router
dhcp	DHCP options. See *SRC-PE Subscribers and Subscriptions Guide, Chapter 6, Classifying Interfaces and Subscribers with the SRC CLI*.
primaryUserName	PPP or DHCP username. This name does not change when the subscriber logs in through a portal.

Configuration Examples for Aggregate Services

For configuration examples for aggregate services see the following guide:

- *SRC-PE Sample Applications Guide*

Configuring Timers for Aggregate Services with the 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;
   ```

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.

Adding an Infrastructure Service with the SRC CLI

To add an infrastructure service:

1. Add the service to be shared, as described in *Adding a Normal Service with the 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;
   ```

Extending Service Implementations with Script Services

You use services to provision policies on a number of systems across a network, including networks that do not contain a JUNOSe 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 JUNOSe routers and JunoScript provisioning on JUNOS routing platforms.

Perform the following to customize service implementations:

1. Writing Scripts for Script Services on page 21

2. Configuring Script Services with the 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 SRC software distribution in the folder *SDK/doc/sae* or in the SAE core API documentation on the Juniper Networks Web site at

http://www.juniper.net/techpubs/software/management/sdx/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 with the 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;
}
```

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;
}
```

To configure a script service:

1. Configure a normal service, but set the service type to script. See *Adding a Normal Service with the 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. Configure the URL of the script service or the path and filename of the service.

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

6. (Optional) Verify your configuration.

   ```
   [edit services scope script service scriptService script]
   user@host# show
   script-type java-class;
   class-name net.juniper.smgt.script.service;
   file://opt/UMC/sae/var/scriptservice/script-service.jar;
   ```

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.

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.

Adding a Mutex Group with the 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 ];
   ```

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 *Chapter 14, Defining and Acquiring Values for Parameters*. 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, 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.

Configuring Service Scopes

The tasks to configure a service scope are:

1. Adding Service Scopes with the SRC CLI on page 28
2. Assigning Services and Mutex Groups to Service Scopes with the SRC CLI on page 29
3. Assigning Service Scopes to VRs or Subscribers with the SRC CLI on page 29

Adding Service Scopes with the 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 with the 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 with the 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
   ```

Service Scope Configuration SRC CLI Examples

The following sections provide two practical examples for using scopes to customize your service configuration.

Example: Delivering 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 *SRC-PE Subscribers and Subscriptions Guide, Chapter 24, Managing Services with Enterprise Manager Portal*.

Example: Customizing 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
   ```

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.

Chapter 2

Managing Services on a Solaris Platform

This chapter describes how to manage services for your SRC configuration with the SRC configuration applications that run only on Solaris platforms. You can also use the SRC CLI that runs on Solaris platforms and the C-series Controller to configure services. See *Chapter 1, Managing Services with the SRC CLI*.

Topics in this chapter include:

Overview of Services

The SRC software supports several types of services:

- Access services—Services that provide access to the Internet or to a content provider's Web site. An access service has the object class *umcAccessService*.
- Outsourced services—Services that wholesalers sell to retailers and that retailers in turn sell to their customers. An outsourced service has the object class *umcOutsourceService*.
- RADIUS services—Services that authenticate subscribers, authorize subscribers' access to the SRC network, and provide accounting information about subscribers' activities (JUNOSe routers only). A RADIUS service has the object class *umcRadiusService*.
- Value-added services (also known as SSP services)—Services that a subscriber pays for in addition to a standard service, such as video on demand, higher bandwidth on demand, e-games, and video conferencing. A value-added service has the object class *sspService*. There are four types of value-added services:
  - Normal—Policy-based service
  - Aggregate—Group of services, handled as a unit
  - Infrastructure—Service that can be activated a number of times across network devices
  - Script—Custom service that is used to provision policies on a number of systems across a network path, including networks that contain network devices that do not have supported network drivers

LDAP Model for Services

You can view service objects in the directory at the distinguished name (DN) *o = Services, o = umc*. If you install the sample data, you can see examples of service configurations through SDX Admin.

For detailed information about the SRC LDAP schema, see the documentation in the SRC software distribution in the folder */SDK/doc/ldap* or on the Juniper Networks Web site at

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

Adding Services

You can add services to the directory with SDX Admin or another LDAP client. The following sections describe how to add each type of service with SDX Admin. For information about using SDX Admin, see *SRC-PE Getting Started Guide, Chapter 43, Using SDX Admin*.

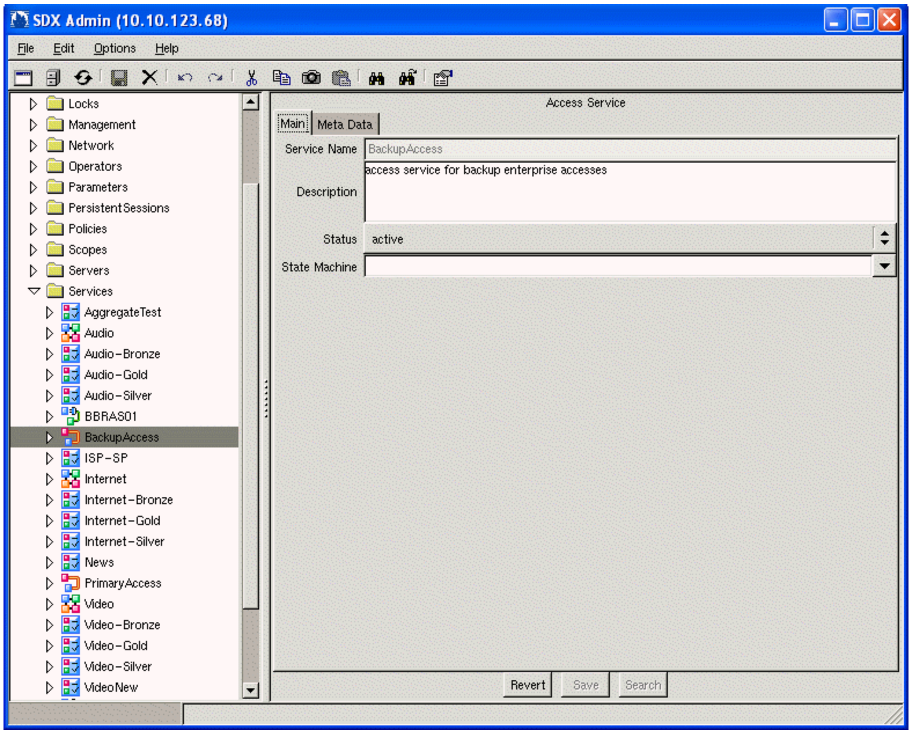
Adding Services to Gain Access to Networks

Access services represent leased line access of an enterprise to the network. To add an access service:

1. In the SDX Admin navigation pane, highlight the **Services** folder, and right-click.
2. Select **New** > **Access Service**.

   The New Access Service dialog box appears.
3. Enter a unique name for the service in the Service Name field, and click **OK**.

   An object for the new service appears in the navigation pane, and basic details for the new service appear in the Main tab of the Access Service pane.



Service Name: BackupAccess

Description: access service for backup enterprise accesses

Status: active

State Machine:

4. Use the field descriptions in *Access Service Fields* on page 36 to configure the service, and then click Save.

Access Service Fields

Use the fields in this section to configure access services.

Description

- Describes the service that subscribers see on a portal application.
- Value—Text
- Default—No value

Status

- Status of this service.
- Value
  - Active—Service accepts new subscriptions.
  - Inactive—Service does not accept new subscriptions.
- Default—No value

State Machine

- DN of a state machine that identifies a set of transitions associated with a workflow for this service. If you specify a DN, all subscriptions to this service should be governed by this state machine.
- Value—Text
- Default—No value

Adding Outsourced Services

To add an outsourced service:

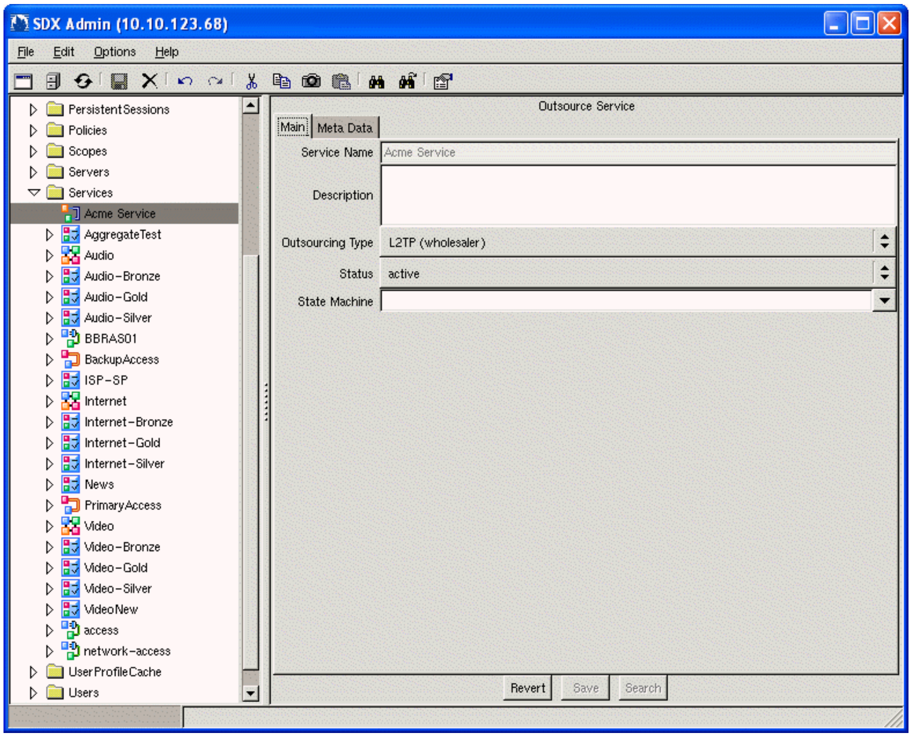
1. In the SDX Admin navigation pane, highlight the **Services** folder, and right-click.
2. Select **New > Outsource Service**.

   The New Outsource Service dialog box appears.
3. Enter the service name in the Service Name field, select the outsourcing type from the drop-down menu, and click **OK**.

4. Enter the service name in the Service Name field, select the type of access service from the Outsourcing Type menu, and click **OK**.

   - Service Name—Unique name of the service
   - Outsourcing Type—One of the following options:
     - L2TP (wholesaler)—Wholesaler manages subscribers and owns equipment that allows subscribers to access SRC services through Layer 2 Tunneling Protocol (L2TP).
     - L2TP (retailer)—Retailer manages subscribers and owns equipment that allows subscribers to access SRC services through L2TP.
     - PTA (wholesaler)—Wholesaler manages subscribers and owns equipment that allows subscribers to access SRC services through PPP Terminated Aggregation (PTA).
     - PTA (retailer)—Retailer manages subscribers and owns equipment that allows subscribers to access SRC services through PTA.

   An object for the new service appears in the navigation pane, and basic details for the new service appear in the Main tab of the Outsource Service pane.



Service Name: Acme Service

Description:

Outsourcing Type: L2TP (wholesaler)

Status: active

State Machine:

5. Use the field descriptions in *Outsourced Service Fields on page 38* to configure the service, and then click **Save**.

Outsourced Service Fields

Use the fields in this section to configure outsourced services.

Description

- Describes the service.
- Value—Text
- Default—No value

Outsourcing Type

- Method that subscribers use to access SRC services and indication of whether the wholesaler or retailer owns the access equipment.
- Value—Option selected in Step 4 on page 37
- Default—No value

Status

- Status of this service.
- Value
  - Active—Service accepts new subscriptions.
  - Inactive—Service does not accept new subscriptions.
- Default—Active

State Machine

- DN of a state machine that identifies a set of transitions associated with a workflow for this service. If you specify a DN, all subscriptions to this service should be governed by this state machine.
- Value—Text
- Default—No value

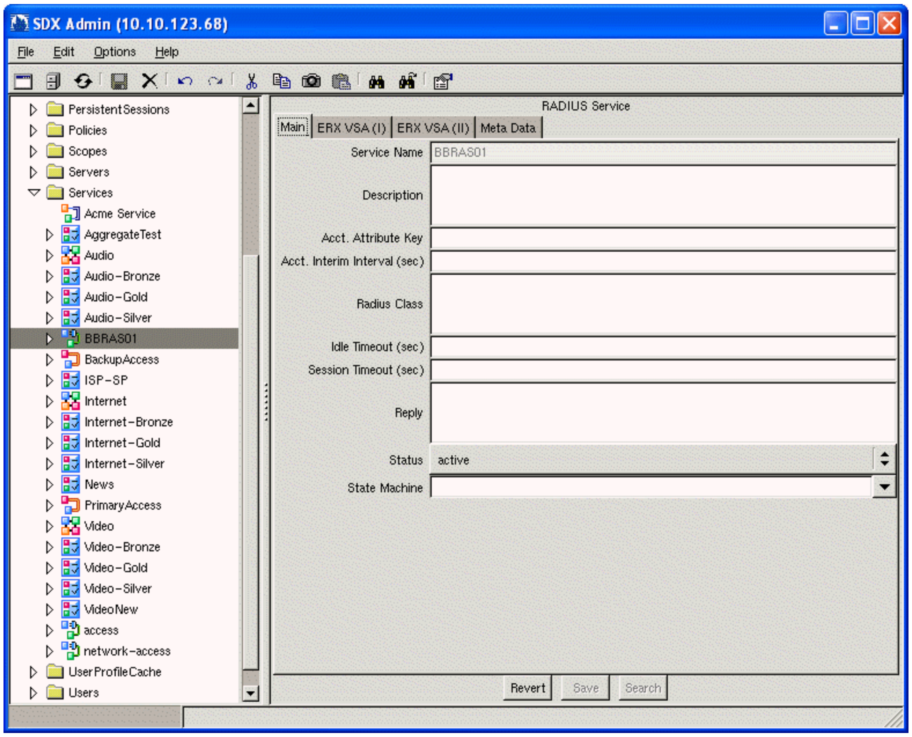
Adding RADIUS Services

JUNOSe routers support the use of RADIUS services; JUNOS routing platforms, however, do not. To add a new RADIUS service:

1. In the SDX Admin navigation pane, highlight the **Services** folder, and right-click.
2. Select **New > RADIUS Service**.

   The New RADIUS Service dialog box appears.
3. Enter a unique name for the RADIUS service name in the Service Name field, and click **OK**.

   An object for the new service appears in the navigation pane, and basic details for the new service appear in the Main tab of the RADIUS Service pane.



Service Name BBRAS01

Status active

4. Use the field descriptions in *RADIUS Service Fields* on page 39 to configure the service, and then click Save.

5. Define how the RADIUS service interacts with the JUNOSe router by filling in the fields in the ERX VSA tabs. See:

   - *Defining Vendor-Specific Attributes in the ERX VSA (I) Tab* on page 41 .
   - *Defining Vendor-Specific Attributes in the ERX VSA (II) Tab* on page 45.

RADIUS Service Fields

Use the fields in this section to configure RADIUS services.

Description

- Describes the service.
- Value—Text
- Default—No value

Acct. Attribute Key

- Identifier that indicates that a subscriber or retailer is billed individually.
- Value—Text
  - For subscribers—Subscriber name
  - For retailers—Domain name
- Default—No value

Acct. Interim Interval (sec)

- Interval between interim accounting messages for this service.
- Value—Number of seconds in the range 0–2147483647
  - No value—The globally configured accounting interim value is used.
  - 0—Interim accounting is disabled for this service.
- Default—No value

Radius Class

- Arbitrary value. If the RADIUS server supplies this value, the network access server (NAS) includes it in all accounting packets for the subscriber.
- Value—Text
- Default—No value

Idle Timeout (sec)

- Time at which the RADIUS session ends if there is no activity between the subscriber and the RADIUS server.
- Value—Number of seconds in the range 0–2147483647
- Default—No value

Session Timeout (sec)

- Time at which the RADIUS session ends.

NOTE: Changes to the session timeout take effect immediately if the new value is lower than the remaining time for a session or if you specify that no session timeout applies. Other changes apply only to services that are activated after you make the change.

- Value—Number of seconds in the range 0–2147483647
- Default—No value

Reply

- Text to be displayed to the subscriber. This is the RADIUS ReplyMessage attribute.
- Value–Text string
- Default–No value

Status

- Status of this service.
- Value
  - Active—Service accepts new subscriptions.
  - Inactive—Service does not accept new subscriptions.
- Default—Active

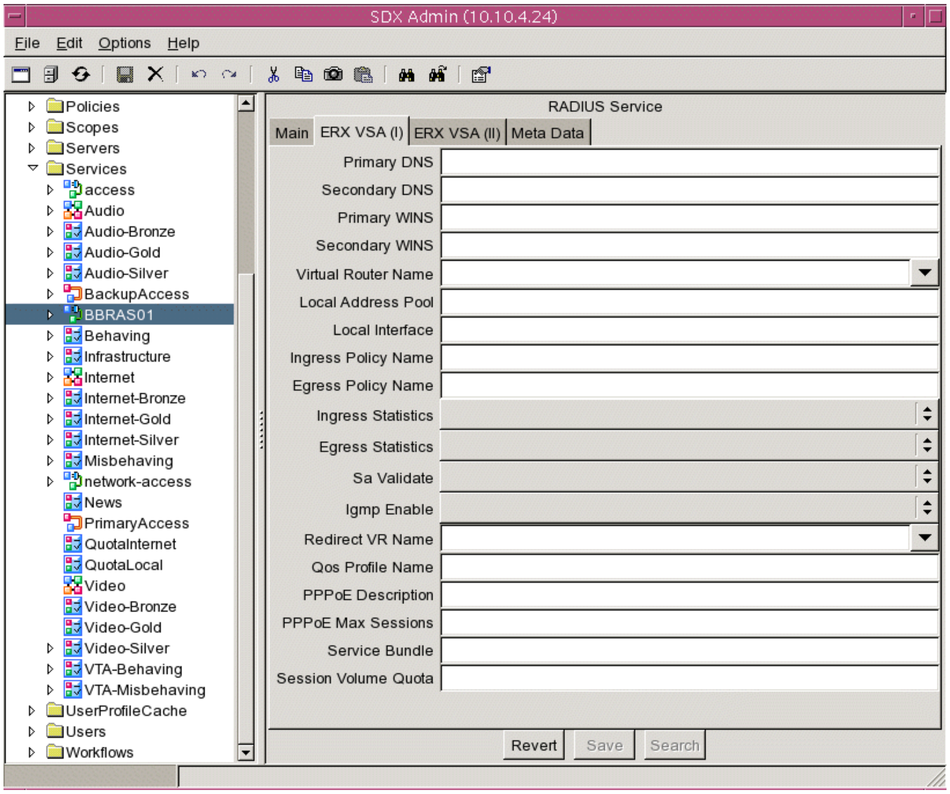
State Machine

- DN of a state machine that identifies a set of transitions associated with a workflow for this service. If you specify a DN, all subscriptions to this service should be governed by this state machine.
- Value—Text
- Default—No value

Defining Vendor-Specific Attributes in the ERX VSA (I) Tab

There are two tabs in the RADIUS service that you can use to enter information about how the RADIUS service interacts: ERX VSA (I) and ERX VSA (II).

You can set the following values in the ERX VSA (I) tab.



Primary DNS

- Subscriber's DNS address negotiated during Internet Protocol Control Protocol (IPCP).
- Value—4-octet IP address
- Default—No value

Secondary DNS

- Subscriber's secondary DNS address negotiated during IPCP.
- Value—4-octet IP address
- Default—No value

Primary WINS

- Subscriber's Windows Internet Naming Service (WINS), also referred to as a NetBIOS Name Server (NBNS), address negotiated during IPCP.
- Value—4-octet IP address
- Default—No value

Secondary WINS

- Subscriber's secondary WINS address negotiated during IPCP.
- Value—4-octet IP address
- Default—No value

Virtual Router Name

- Name of the virtual router (VR) through which subscribers can access this RADIUS service.
- Value
  - blank—VR is not selected
  - default—Default VR
  - <vrName>—Name of VR on which PPP interface is created
- Default—No value

Local Address Pool

- Name of a local address pool from which a VR assigns IP addresses.
- Value—Text
- Default—No value

Local Interface

- Interface on a JUNOSe router.
- Value—Text
- Default—No value

Ingress Policy Name

- Name of an input policy to apply to the subscriber's interface.
- Value—Text
- Default—No value

Egress Policy Name

- Name of an output policy to apply to the subscriber's interface.
- Value—Text
- Default—No value

Ingress Statistics

- Indicates whether ingress statistics are generated on the subscriber's interface.
- Value
  - blank—Router uses default setting
  - disable—Disables generation of statistics
  - enable—Enables generation of statistics
- Default—Blank

Egress Statistics

- Indicates whether egress statistics are generated on the subscriber's interface.
- Value
  - blank—Router uses default setting
  - disable—Disables generation of statistics
  - enable—Enables generation of statistics
- Default—Blank

Sa Validate

- Specifies whether the source address on the subscriber's interface is validated.
- Value
  - blank—Router uses default setting
  - disable—Disables validation
  - enable—Enables validation
- Default—Blank

IGMP Enable

- Specifies whether the subscriber can register to receive multicast services through Internet Group Management Protocol (IGMP).
- Value
  - blank—Router uses default setting
  - disable—Disables IGMP
  - enable—Enables IGMP
- Default—Blank

Redirect VR Name

- VR name that identifies the VR context in which to authenticate the subscriber.
- Value—Text
- Default—No value

QoS Profile Name

- Name of the quality of service (QoS) profile to attach to the subscriber's interface.
- Value—Text
- Default—No value

PPPoE Description

- String pppoe<mac addr> that the router obtains from Point-to-Point Protocol over Ethernet (PPPoE) operations and sends to the RADIUS server.
- Value—Text
- Default—No value

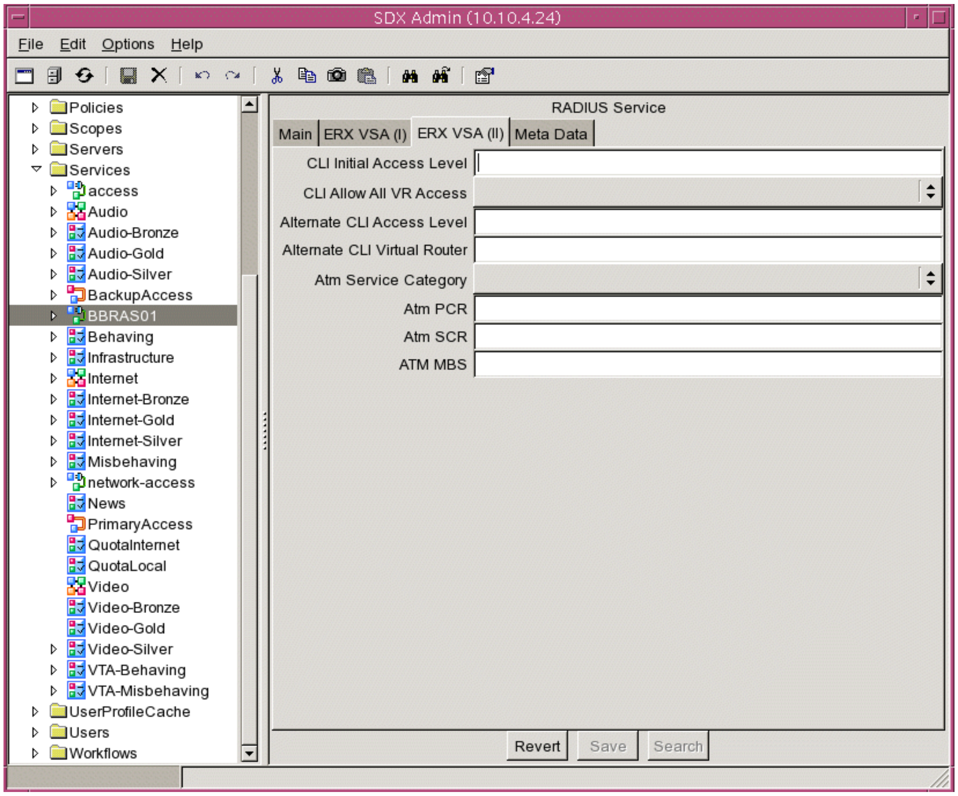
Service Bundle

- SRC service bundle.
- Value—Text
- Default—No value

Defining Vendor-Specific Attributes in the ERX VSA (II) Tab

There are two tabs in the RADIUS service that you can use to enter information about how the RADIUS service interacts: ERX VSA (I) and ERX VSA (II).

You can set the following values in the ERX VSA (II) tab.



CLI Initial Access Level

- Privilege level for the JUNOSe command-line interface (CLI) that determines the command to which subscribers of this RADIUS service have access.

  See the *JUNOSe System Basics Configuration Guide* for information about security and the JUNOSe CLI.

- Value—Text
- Default—No value

CLI Allow All VR Access

- Specifies which VRs subscribers can access.
- Value
  - blank—Router uses default setting.
  - disable—Subscribers can access only the specified VRs.
  - enable—Subscribers can access all VRs.
- Default—Blank

Alternate CLI Access Level

- Secondary (backup) level of access to the CLI.
- Value—Text
- Default—No value

Alternate CLI Virtual Router

- Name of a secondary (backup) VR associated with this RADIUS service.
- Value—Text
- Default—No value

Atm Service Category

- Asynchronous transfer mode (ATM) traffic management rate.
- Value
  - blank—Router uses default setting
  - UBR—Unspecified bit rate (UBR)
  - UBRPCR—UBR with a peak cell rate (PCR)
  - nrtVBR—Variable bit rate, non–real time (VBR-NRT)
  - CBR—Constant bit rate (CBR)
- Default—No value

Atm PCR

- Peak cell rate (PCR).
- Value—4-octet integer
- Default—No value

Atm SCR

- Sustained cell rate (SCR).
- Value—4-octet integer
- Default—No value

ATM MBS

- Maximum burst rate (MBS).
- Value—4-octet integer
- Default—No value

Adding Value-Added Services

A value-added service is one that subscribers activate and deactivate. The SAE supports the following types of value-added services:

- Normal—Policy-based service
- Aggregate—Group of services, handled as a unit
- Infrastructure—Service that can be activated a number of times across network devices
- Script—Custom service that is used to provision policies on a number of systems across a network path, including networks that contain network devices that do not have supported network drivers

Use aggregate and infrastructure services together to apply policies across JUNOSe routers and JUNOS routing platforms, and other systems that have a supported router driver. Use script services to create customized service implementations, such as a configuration to provision policies for a Multiprotocol Label Switching (MPLS) tunnel. Script services can be used with aggregate and infrastructure services to provide a custom implementation across network devices, some of which do not have a supported router driver.

Before You Configure Value-Added 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 *Defining Policies to Manage Traffic* on page 139.

NOTE: If you add more value-added services than your license supports, the software logs and publishes errors through SNMP traps, and the SAE may shut down. If you have defined more value-added services than you require, you can resolve the situation by deleting some value-added services or setting their deleted LDAP attributes to true. In the latter case, the SAE cannot use the services; however, they will still exist in the directory.

Adding a Normal Value-Added Service

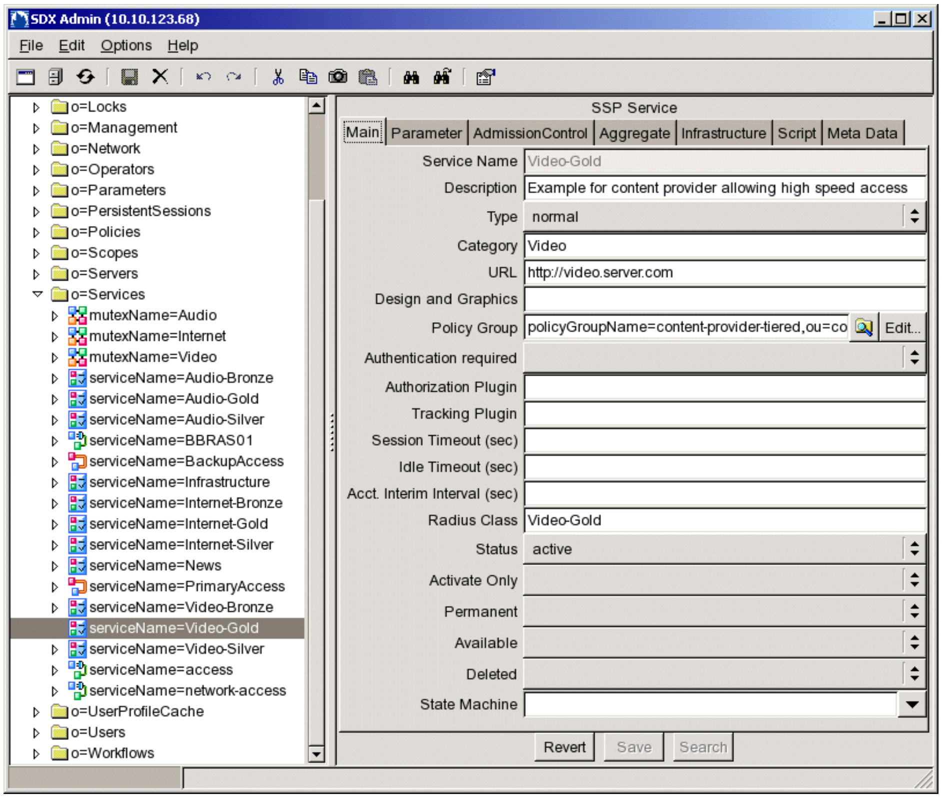
To add a normal value-added service with SDX Admin:

1. In the SDX Admin navigation pane, highlight the **Services** folder, right-click, highlight **New**, and then click **SSP Service**.

   The New SSP Service dialog box appears.

2. Enter a unique name for the SSP service name in the Service Name field, and click **OK**.

   An object for the new service appears in the navigation pane, and basic information about the new service appears in the Main tab of the SSP Service pane.



Service Name: Video-Gold
Description: Example for content provider allowing high speed access
Type: normal
Category: Video
URL: http://video.server.com
Design and Graphics:
Policy Group: policyGroupName=content-provider-tiered,ou=co
Authentication required:
Authorization Plugin:
Tracking Plugin:
Session Timeout (sec):
Idle Timeout (sec):
Acct. Interim Interval (sec):
Radius Class: Video-Gold
Status: active
Activate Only:
Permanent:
Available:
Deleted:
State Machine:

3. In the Main tab, set **Type** to normal.

4. Use the field descriptions in *Value-Added Service Fields* on page 49 to configure the service, and then click **Save**.

5. (Optional) You can configure parameters for the value-added service. See *Setting Parameters for Value-Added Services* on page 53.

Value-Added Service Fields

Use the fields in this section to configure normal value-added services.

Description

- Describes the service that subscribers see on the portal.
- Value—Text
- Default—No value

Type

- Type of service.
- Value
  - normal—Individual service that a subscriber activates and deactivates
  - aggregate—Group of normal services that a subscriber activates and deactivates as a unit

    For information about aggregate services, see *Aggregating Services* on page 59.
  - script—Custom service that is used to provision policies on a number of systems across a network path, including networks that contain network devices that do not have supported network drivers
  - infrastructure—Service that can be activated a number of times across network devices
- Default—Normal

Category

- Text that appears in the set of tabs that categorize services in the residential portal; for example, Video.
- Value—Text
- Default—No value

URL

- URL of the Web page that the subscriber sees after activating a service.
- Value—Text
- Default—No value

Design and Graphics

- Text string in the directory when the service is defined. The portal pages can use this string for any purpose.

  The portal pages retrieve this string from the appropriate service object and incorporate the string in a URL that points to a file or directory that contains service-specific items, such as GIF files and Web pages. As a consequence, portal pages can be customized according to the available services that a subscriber has activated.
- Value—Text
- Default—No value

Policy Group

- DN of the policy group that is applied when the service is activated. The policy engine does not allow the activation of a service without an associated policy group.

  If you do not have a policy group defined for this service, define a policy group with an empty ingress policy list and an empty egress policy list, and attach it to the service. See *Defining Policies to Manage Traffic* on page 139 for details.

  Applies only to normal services.
- Value—Text
- Default—No value

Authentication Required

- Determines whether activation of this service requires authentication with a username and password that are specific to this service.
- Value
  - blank—Default (false)
  - true—Authentication required
  - false—Authentication not required
- Default—Blank

Authorization Plugin

- List of authentication plug-ins that are called before the service is activated. In the list, a comma separates each authentication plug-in from the next one in the list.
- Value—Text
- Guidelines—If you use an authorization plug-in and define schedules for services, add the configured schedule authorization plug-in to the list. The default name for a schedule authorization plug-in is scheduleAuth.
- Default—No value

Tracking Plugin

- List of tracking plug-ins that are called after the service is activated, during interim updates, and when the service has been deactivated. In the list, a comma separates each authentication plug-in from the next one in the list.
- Value—Text
- Default—No value

Session Timeout (sec)

- Time after which the service session is deactivated.
- Value—Number of seconds in the range 0–2147483647
- Guidelines—Changes to the session timeout take effect immediately if the new value is lower than the remaining time for a session or if you specify that no session timeout applies. Other changes apply only to services that are activated after you make the change.
- Default—No value

Idle Timeout (sec)

- Time that a service is idle, after which the SAE deactivates the service.

  To decide whether a service is idle, the SAE collects accounting information for the service, which means that the service activation policy must specify an accounting rule. The idle timeout is the minimum time the service must be idle before it is deactivated; the actual deactivation can be up to the accounting interim interval.
- Value—Number of seconds in the range 0–2147483647
- Default—No value

Acct. Interim Interval (sec)

- Time between interim accounting messages for this service.
- Value—Number of seconds in the range 0–2147483647
  - blank—Uses the globally configured accounting interim value
  - 0—Disables interim accounting for this service
- Default—No value

Radius Class

- Default value used in the RADIUS class attribute in RADIUS accounting messages. If RADIUS authenticates the service session, the class attribute received in the RADIUS Access-Accept response from the server overrides this value.
- Value—Text
- Default—Service name

Status

- Specifies whether this service is active.
- Value
  - Active—The service is available for new subscriptions.
  - Inactive—No new subscriptions are accepted.
- Default—Active

Activate Only

- Determines whether the SAE can deactivate this service.
- Value
  - blank—False.
  - true—SAE can activate but not deactivate this service.
  - false—SAE can activate and deactivate this service.
- Default—Blank

Permanent

- Determines whether the SAE maintains permanent activation of this service for a subscriber.
- Value
  - blank—False.
  - true—SAE activates this service automatically when a subscriber with a subscription to this service logs in, and keeps this service active until the subscriber logs out.
  - false—SAE can activate and deactivate this service based on subscribers' requests.
- Default—Blank

Available

- Determines whether a subscriber can activate a service.
- Value
  - blank—True.

CAUTION: Do not use the default (blank) setting for this field; the directory may not operate correctly if you do.

  - true—Subscriber can activate service.
  - false—Subscriber cannot activate service.
- Default—Blank

Deleted

- Specifies the availability of this entry to other SRC components connected to the directory.
- Value
  - Blank—Other SRC components can access this entry in the directory.
  - True—Other SRC components cannot use this entry in the directory, although the object still exists.
  - False—Other SRC components can access this entry in the directory.
- Default—Blank

State Machine

- DN of a state machine that identifies a set of transitions associated with a workflow for this service. If you specify a DN, all subscriptions to this service should be governed by this state machine.
- Value— < DN of the state machine >
- Default—No value

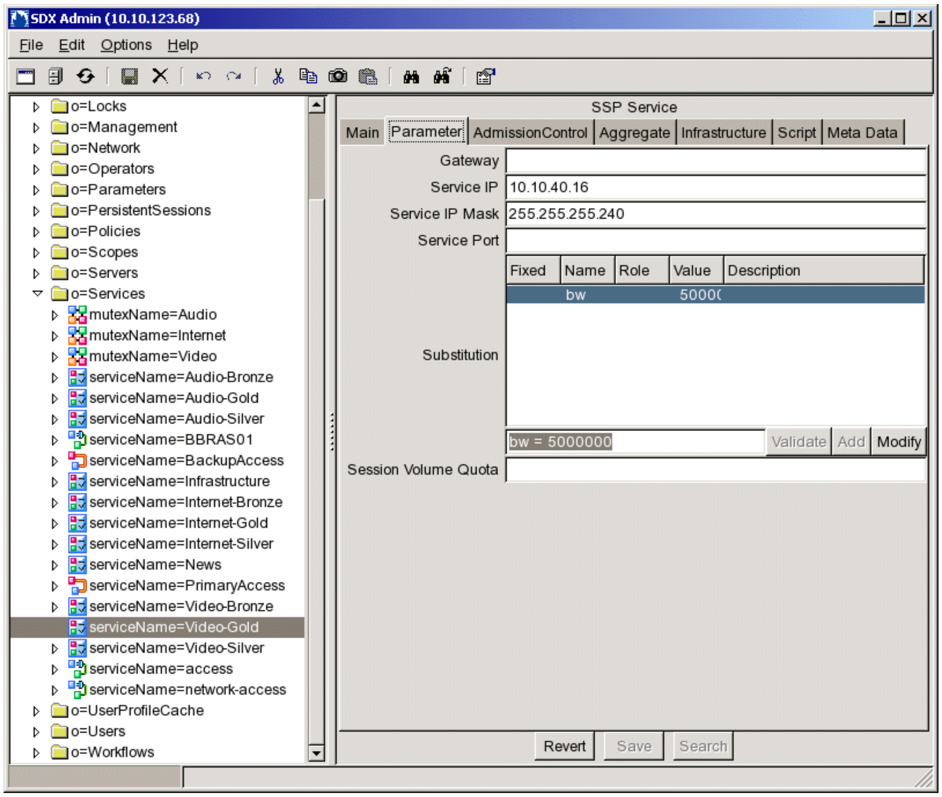
Setting Parameters for Value-Added Services

Using parameters, you can define general settings in one 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 *Chapter 14, Defining and Acquiring Values for Parameters*.

To configure parameters for value-added services:

1. In the SDX Admin navigation pane, select a value-added service, and then click the **Parameter** tab.

   The Parameter tab appears in the content pane.



Gateway:

Service IP: 10.10.40.16

Service IP Mask: 255.255.255.240

Service Port:

Substitution:

Fixed	Name	Role	Value	Description
	bw		5000(	

bw = 5000000

Session Volume Quota:

2. Use the field descriptions in *Parameter Fields* to configure parameters for value-added services.

Parameter Fields

Use the fields in this section to configure parameters for value-added services.

Gateway

- Actual IP address of the gateway router. This value is substituted for the policy global parameter called gateway_ipAddress.
- Value—<IP address>
- Default—No value

Service IP

- Actual IP address of the host(s) that provides the service. This value is substituted for the policy global parameter called service_ipAddress.
- Value—<IP address>
- Guidelines—This entry is needed only if the policy group in the service is referencing this parameter.
- Default—No value

Service IP Mask

- Actual IP mask for the service. This value is substituted for the policy global parameter called service_ipMask.
- Value—<IP mask>
- Guidelines—This entry is needed only if the policy group in the service is referencing this parameter.
- Default—No value

Service Port

- Actual Transmission Control Protocol (TCP)/User Datagram Protocol (UDP) port for the service. This value is substituted for the policy global parameter called service_port.
- Value—<port number>
- Default—No value

Substitution

- Substitutions for other parameters (see *Configuring Substitutions* on page 56).
- Value—<substitution in correct syntax>
- Default—No value

Session Volume Quota

- 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.
- Value—<downstream Quota>.<upstreamQuota>
  - <downstream Quota>—Number of bytes available for transmitting data from the network to the subscriber
  - <upstreamQuota>—Number of bytes available for transmitting data from the subscriber to the network

- Guidelines—The value of a service session can be defined at runtime either through an authorization plug-in or a call to the SAE API.

  If the Session Volume Quota attribute is defined in more than one place, which value is used depends on where the value is defined. The SRC software searches for the value in the following order:

  1. Value set in a call to the SAE

  2. Value set in an authorization

  3. Value set in a service definition

- Default—No default

Configuring Substitutions

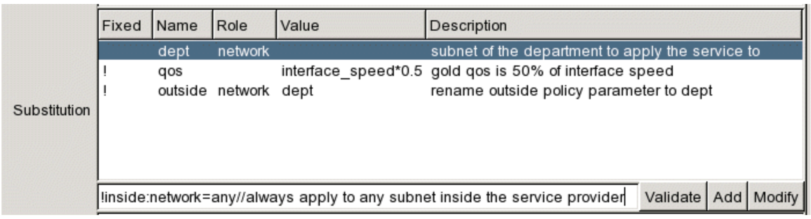
This section shows how to add, modify, validate, and delete substitutions in SDX Admin.

Adding Substitutions

To add a substitution:

1. In SDX Admin, select the **Parameter** tab for the service to which you want to add a substitution.

2. In the unlabeled field below the Substitution field, enter the substitution in the correct syntax (see *Formatting Substitutions* on page 403*)*. For example:



Fixed	Name	Role	Value	Description
	dept	network		subnet of the department to apply the service to
!	qos		interface_speed*0.5	gold qos is 50% of interface speed
!	outside	network	dept	rename outside policy parameter to dept

!inside:network=any//always apply to any subnet inside the service provider

3. Click **Add**.

NOTE: Substitutions for JUNOSe routers may not correctly display in the Substitution field for SDX Admin. To confirm the syntax of a JUNOSe substitution, click on the substitution in the Substitution field, and observe the syntax in the entry field below the Substitution field.

Substitutions to a Transmission Rate for a Scheduled Action

When you use SDX Admin to assign substitutions to the Transmit Rate Unit for a Scheduler action, you can specify one of the following:

- “percent”
- “remainder”
- “bps”

Do not use the “rate_in_percent” value as it appears in Policy Editor for substitutions in SDX Admin. Do one or the other. For example in Policy Editor, specify a parameter called ‘x’ for the Transmit Rate Unit for a Scheduler Action and select rate_in_percent; or in SDX Admin, create a substitution as x = percent.

Modifying Substitutions

To modify a substitution:

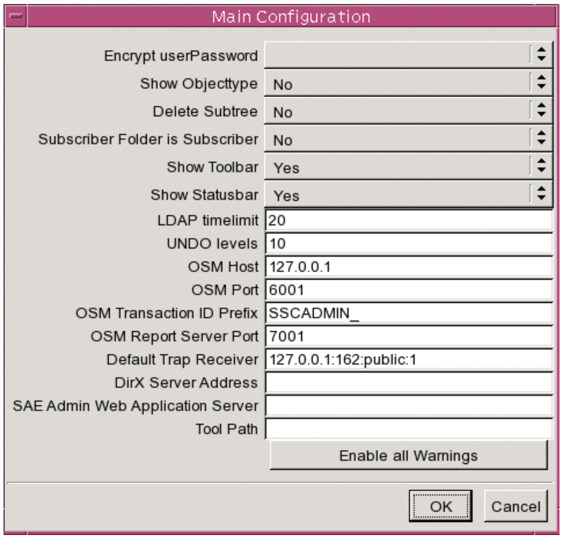
1. In SDX Admin, select the **Parameter** tab for the service to which you want to add a substitution.
2. Select the substitution in the Substitutions field.
3. Modify the substitution in the unlabeled field below the Substitution field.
4. Click **Modify**.

Validating Substitutions

To validate a substitution:

1. In SDX Admin, select **Options** > **Configure**.

   The Main Configuration window appears.



Encrypt userPassword	
Show Objecttype	No
Delete Subtree	No
Subscriber Folder is Subscriber	No
Show Toolbar	Yes
Show Statusbar	Yes
LDAP timelimit	20
UNDO levels	10
OSM Host	127.0.0.1
OSM Port	6001
OSM Transaction ID Prefix	SSCADMIN_
OSM Report Server Port	7001
Default Trap Receiver	127.0.0.1:162:public:1
DirX Server Address	
SAE Admin Web Application Server	
Tool Path	

2. In the SAE Admin Web Application Server field, enter the identifier of the host on which you installed SAE Web Admin, in the format: <host>:<port>.

   - <host>—Name or IP address of the host
   - <port>—Port number for SAE Web Admin

3. Click **OK**.

4. Select the substitution in the Substitution field.

5. Click **Validate**.

   SDX Admin displays the result of the validation.

Deleting Substitutions

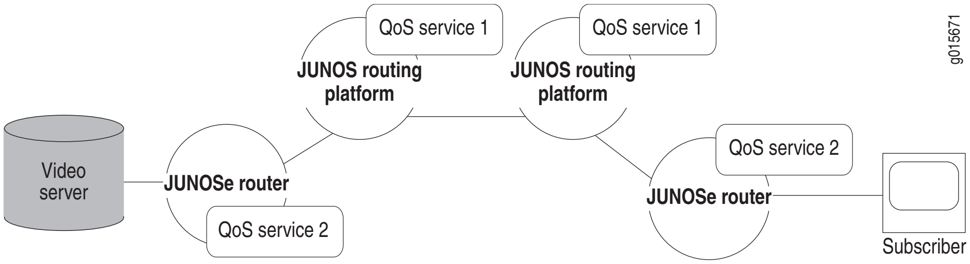
To delete a substitution, select it in the Substitutions field, right-click, and select **Delete**.

Aggregating Services

An aggregate service is a type of value-added service that comprises a number of individual 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 3 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 3: 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 comprise 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, DN, object path, login name, or associated virtual router.

To use aggregate services requires that the 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 OnPopDnSharedIp, 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 *SRC-PE Network Guide: SAE, Juniper Networks Routers, NIC, and SRC-ACP, Chapter 19, 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.

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.

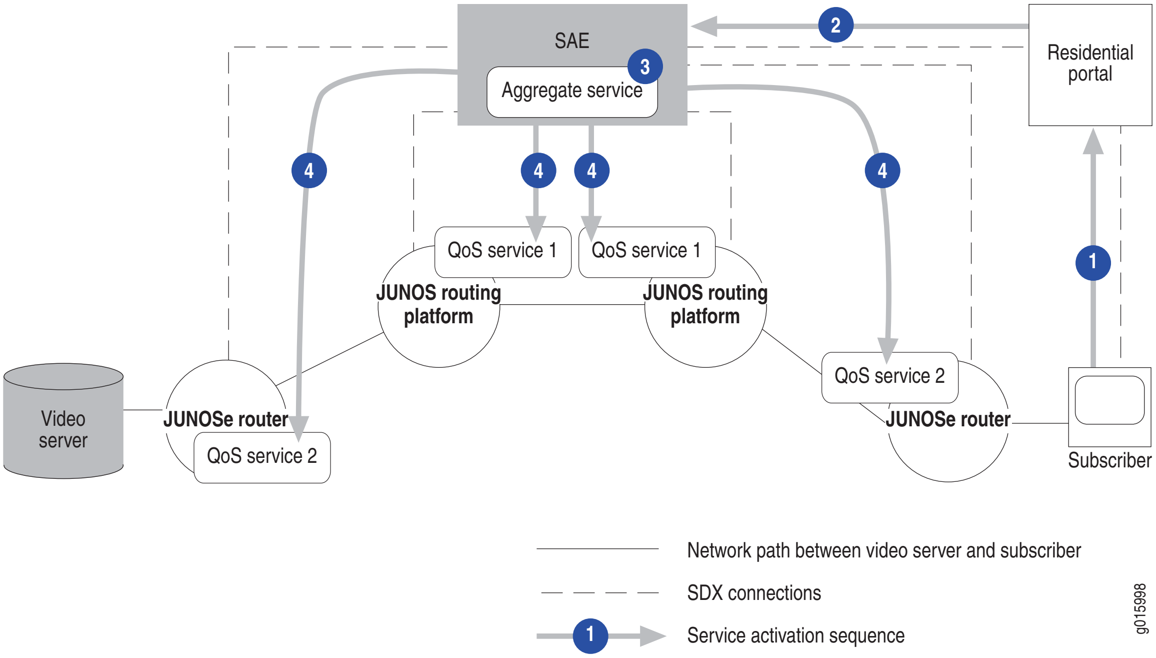
Session Monitoring

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 COPS client stops on a JUNOSe 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 similar way as any other value-added service, but with the additional requirement of activating the associated fragment services. Figure 4 shows a sample service activation for a video-on-demand service.

Figure 4: Aggregate Service Activation



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

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.
5. 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.
6. The aggregate service initiates accounting, if 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.

For detailed information about service activation, see *SRC-PE Subscribers and Subscriptions Guide, Chapter 3, Subscriber Logins and Service Activation*.

Before You Configure an Aggregate Service

Before you configure an aggregate service:

1. Plan the aggregate services:

   - 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 Value-Added Service* on page 48.

3. If the aggregate service includes services to be activated remotely, ensure that the NIC is configured and running on each SAE that resides in your SRC network.

4. Ensure that the NIC is configured to use a scenario that provides the appropriate type of key.

   See *Subscriber Reference Expressions for Fragment Services* on page 59.

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 *SRC-PE Getting Started Guide, Chapter 34, Defining an Initial Configuration on a Solaris Platform*.

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.

   See *Reviewing Service Status* on page 89.

Adding an Aggregate Service

To use SDX Admin to add an aggregate service:

1. In the navigation pane, right-click the **Services** folder, select **New**, and then select **SSP Service**.

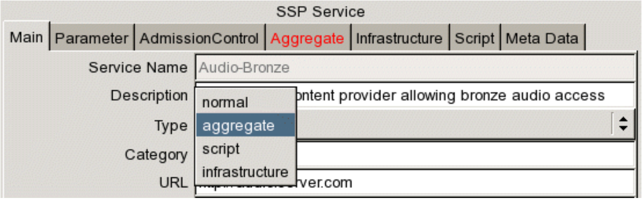
   The New SSP Service dialog box appears.

2. Enter a unique name for the SSP service name in the Service Name field, and click **OK**.

   An object for the new service appears in the navigation pane, and basic information about the new service appears in the Main tab of the SSP Service pane.

3. In the Main tab, specify values for the following fields:

   - Description—Description of the service
   - Type—Aggregate

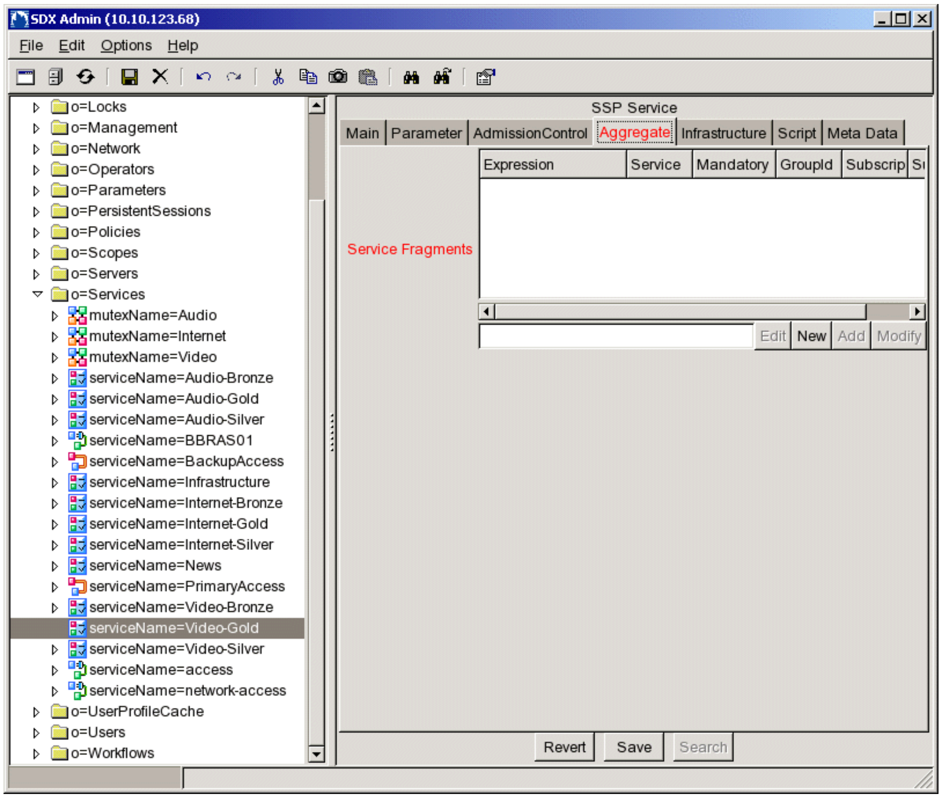


Service Name Audio-Bronze
Description ontent provider allowing bronze audio access
Type aggregate
Category
URL rver.com

If you want to specify values for other fields in the Main tab, see *Value-Added Service Fields* on page 49.

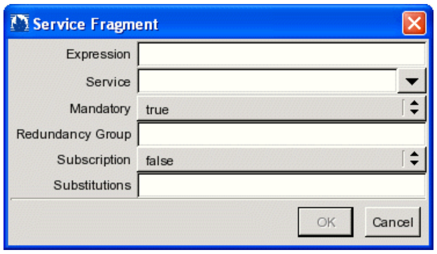
4. Click the **Aggregate** tab.

   The Aggregate tab appears in the content pane.



5. In the Aggregate tab, click **New** to define a fragment service to be included in the aggregate service.

   The Service Fragment dialog box appears.


Service Fragment

Expression	
Service	
Mandatory	true
Redundancy Group	
Subscription	false
Substitutions	

6. Edit the values in the Service Fragment dialog box, and then click **OK**.

   See *Service Fragment Fields* on page 66.

7. In the Aggregate Service tab, click **Add** to add the fragment service to the aggregate service.

8. Repeat Steps 5–7 for each fragment service to be added to the aggregate service.

9. Click **Save**.

Configuration Examples for Aggregate Services

For configuration examples for aggregate services see the following guide:

- *SRC-PE Sample Applications Guide*

Service Fragment Fields

In SDX Admin, you can modify the fields in this section to configure a fragment service for an aggregate service in the Service Fragment dialog box.

Expression

- Subscriber reference expression that identifies the remote subscriber session that will host the fragment. The remote subscriber session is an assigned IP subscriber. If the remote SAE manages the specified interface, the SAE creates an assigned IP subscriber session if necessary.
- Value—Use one of the following values to identify the remote subscriber session. The items in the list show the syntax to use.
  - current—The remote subscriber session is the same as the current subscriber session
  - address = "<IP address>"
  - vr = "<virtual-router name>", interfaceName = "<interface name>"
  - vr = "<virtual-router name>", ifIndex = "<interface index>"
  - dn = "<DN of the subscriber profile>"
  - vr = "<virtual router name>", interfaceName = "<interface name>", address = "<IP address>"
  - login_name = "<login-name>"
  - vr = "<virtual-router name>", login_name = "<login-name>"
  - primary_user_name = "<PPP login name | authenticated DHCP login name>"

  - vr = "<virtual-router name>", primary_user_name = "<PPP login name | authenticated DHCP login name>"
  - ref= "<path>"

    The <path> identifies the hierarchy of directory objects below the LDAP object o=aggregateService. The final object contains the attribute subscriberRefExpr to identify the subscriber session. A forward slash (/) separates the objects in the path.

- Guidelines—You can also use Python expressions to specify literal values listed above. Python expressions access and manipulate data in a subscriber session and a service session, including substitutions. For example, to use a Python expression for a substitution, type <- before the expression and -> after it; for example, <-ifAlias->.

  For information about substitutions, see *Configuring Substitutions* on page 56.

  For information about using Python expressions to represent values in a subscriber reference expression, see *Using Python Expressions in a Subscriber Reference Expression* on page 69.

  To create Python expressions, use the fields in Table 6 on page 69. You can specify more than one string in a Python script expression.
- Default—No value
- Examples
  - current
  - address="10.10.10.1"
  - vr="<-substitution.serviceVr->",interfaceName= "<-substitution.serviceInterface->"
  - dn = "uniqueId=<-ifAlias->,<-userDn->"
  - vr=<-["vr1","vr2"]->,loginName=<-["joe","jane"]->

Service

- Value-added service to be included in the aggregate service as a fragment service.
- Value—Menu of value-added services that have already been configured
- Default—No value

Mandatory

- Specifies whether the fragment service must be active for the aggregate service to become active.
- Value
  - mandatory—Fragment service must be active for the aggregate service to become active.
  - optional—Fragment service does not need to be active for the aggregate service to become active.
- Default—Mandatory

Redundancy Group

- Group name to be applied to each fragment service that is to be part of a redundancy group. The fragment services that have the same group name provide redundancy for each other.
- Value—Text
- Default—No value

Subscription

- Specifies whether a remote subscriber session is required to subscribe to the fragment service.
- Value
  - true—Remote subscriber session must be subscribed to the fragment service for it to become active.
  - false—Remote subscriber session does not need to be subscribed to the fragment service for it to become active.
- Guidelines—Enabling subscription can be used to limit the services that can be activated as fragments.

  Setting this field to true lets you control which services can be used as fragments. For example, for an aggregate service that supports VoIP to push a policy to the caller and the callee, you can require that both subscribers sign up for VoIP services. If you set the field to false, only one party needs to subscribe to the aggregate service; the policy service sessions are created automatically.
- Default—False

Substitutions

- List of substitutions, in the correct syntax, that are used as arguments for the fragment to become active. If a parameter does not acquire a value, the associated fragment service does not become active.

  For information about acquiring substitution values, see *Chapter 14, Defining and Acquiring Values for Parameters*.
- Value
  - <parameter-name>
  - <parameter-name> = <substitution-expression>

  Use commas to separate multiple substitutions.
- Guidelines—If you specify <parameter-name> for the value, the parameter name is defined to have the same value in the fragment service session as in the aggregate service session.
- If you specify <parameter-name> = <substitution-expression> for the value, the parameter name on the left side of the equals sign is defined for the fragment service session. This parameter name is the result of the evaluation of the expression (in the aggregate service session) on the right side of the equals sign.
- Default—No value

Using Python Expressions in a Subscriber Reference Expression

You can compose Python expressions from one or more of the fields in Table 6 for the definition of a subscriber reference expression of a fragment service. You enter these expressions in the Expression field of the Fragment Service dialog box in which you define a fragment service for an aggregate service.

For information about configuring fragment services for an aggregate service, see *Adding an Aggregate Service* on page 64.

Table 6: Fields Used in Python Expressions for Aggregate Services

Field	Description
substitution.<xyz>	Value of the substitution <xyz>. Substitutions are acquired by means of the regular acquisition path for service sessions. The name of substitutions is 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.
loginType	The type of subscriber session, one of the following: ■ 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 AUTHINTF 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 ADDR login is triggered when the DHCP server in the JUNOSe router provides a token IP address. (JUNOSe routers) ■ AUTHADDR—An AUTHADDR 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

Table 6: Fields Used in Python Expressions for Aggregate Services (continued)

Field	Description
ifDesc	Alternate name for the interface. This is the name used by the Simple Network Management Protocol (SNMP). On a JUNOSe router the format of the description is: ip < slot > / < port > . < subinterface > On a JUNOS routing platform, ifDesc is the same as interfaceName.
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
nasIp	NAS IP address of the router
dhcp	DHCP options. See *SRC-PE Subscribers and Subscriptions Guide, Chapter 7, Classifying Interfaces and Subscribers on a Solaris Platform*.
primaryUserName	The PPP or DHCP username. This name does not change when the subscriber logs in through a portal.

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 a subscriber 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.

Adding an Infrastructure Service

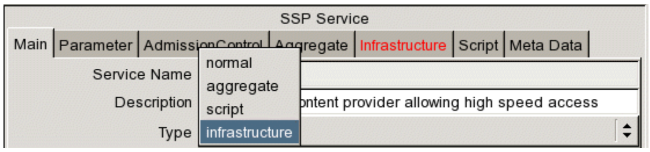
To add an infrastructure service:

1. Configure the value-added service to be shared, or identify an existing value-added service to share.

   This service can be any type of service except an aggregate service.

2. In the SDX Admin navigation pane, right-click the **Services** folder, highlight **New**, and then click **SSP service**.

3. In the Main tab page, select **infrastructure** in the Type field.



SSP Service

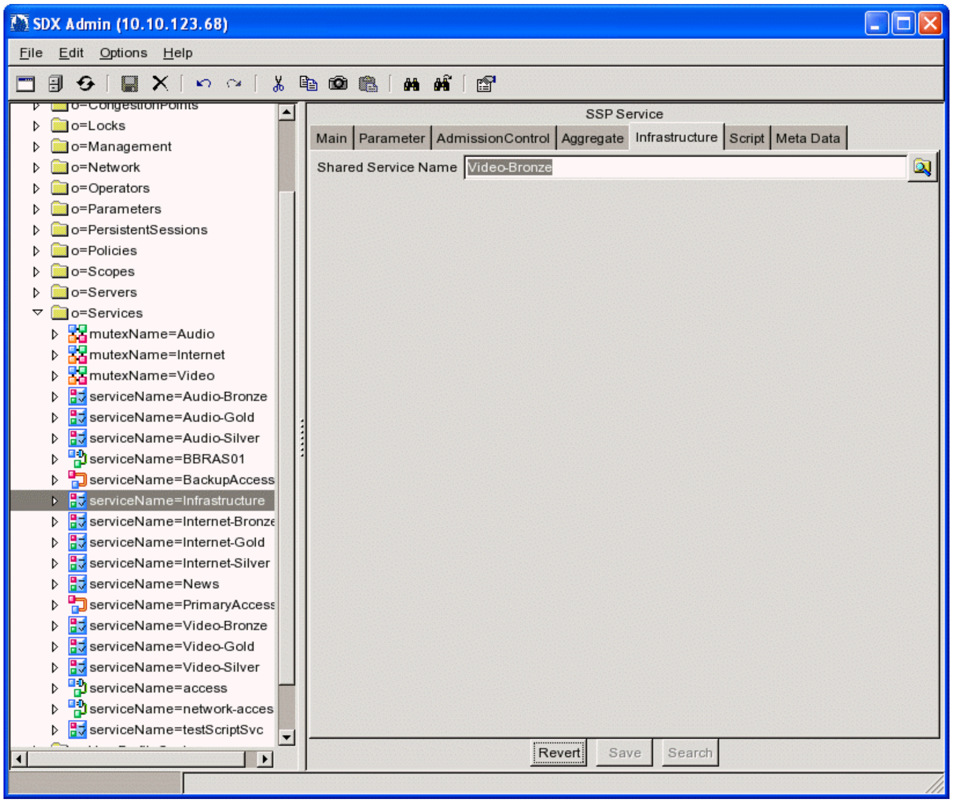
Service Name

Description ontent provider allowing high speed access

Type infrastructure

4. Click the **Infrastructure** tab.

   The Infrastructure tab appears in the content pane.



SDX Admin (10.10.123.68)

SSP Service

Shared Service Name Video-Bronze

5. Select the service to be shared in the Shared Service Name field, and click **Save**.

Extending Service Implementations with Script Services

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

- Provisioning of layer 2 devices, such as digital subscriber line access multiplexers (DSLAMs).
- Setting up of network connections such as MPLS tunnels.
- Provisioning of policies for network devices that do not have a supported SAE router driver.

To customize service implementations:

1. Write a script that implements the ScriptService interface, a service provider interface (SPI) for the SAE.
2. Add a script service that references the script.

Writing Scripts for Script Services

The ScriptService 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 SRC software distribution in the folder *SDK/doc/sae* or in the SAE core API documentation on the Juniper Networks Web site at

http://www.juniper.net/techpubs/software/management/sdx/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 passive; that is, 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, in 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 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: 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: 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() {
    }
}
```

Adding Script Services

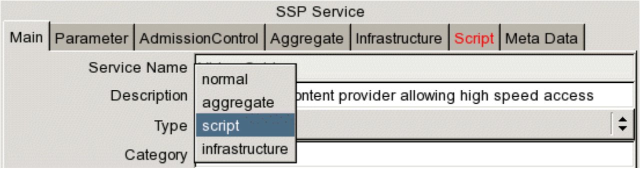
Before you add a script service, make sure that you know the location of the script file that the service will reference.

To add a script service:

1. In the SDX Admin navigation pane, right-click the **Services** folder, highlight **New**, and then click **SSP service**.

2. Enter a name for the service, and click **OK**.

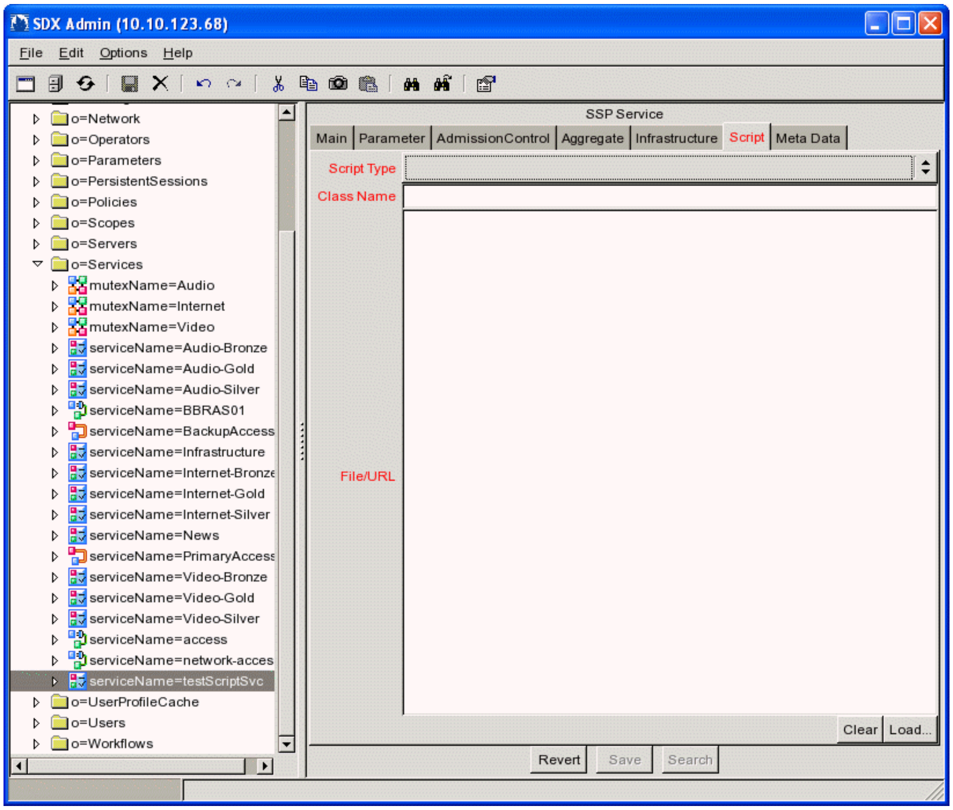
3. In the Main tab pane, select **Script** in the Type field.



Description: ntent provider allowing high speed access

4. Click the **Script** tab.

   The Script tab appears in the content pane.



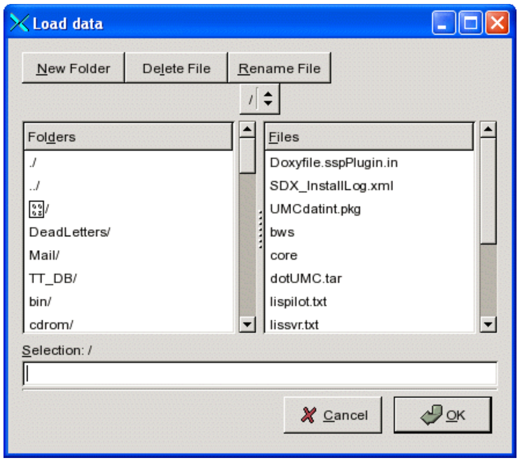
5. Using the field descriptions in *Configuring Values for Script Services* on page 77, configure the Script Type and Class Name.

6. If the script type is URL, enter the URL in the File/URL box.

   or

   If the script type is not URL, click **Load**.

   The Load data dialog box appears.



7. Select the directory that holds the script that contains the implementation of the ScriptService interface; then select the file. Or type the path to the script file in the Selection box, and click **OK**.

   If a JAVA implementation includes more than one JAR file, use commas to separate file URL entries or enter one URL per line.

   The content of the script file appears in the File/URL box in the Script pane.

You can manipulate files and folders from the Load data dialog box.

- To create a new folder, click **New Folder**.
- To remove a file, click **Delete File**.
- To rename a file:
  1. In the Files list, select a file, and click **Rename File**.

     The Rename File dialog box appears.

  2. Enter the new filename, and click **OK**.

Configuring Values for Script Services

Use the following field descriptions to provide information about the script to be used by the script service.

Script Type

- Type of script that the script service uses.
- Value
  - URL—URL to identify the location of script file
  - Python—Jython source code
  - Java Class—Compiled Java class file
  - Java Archive—Java archive file (.jar)
- Default—No value

Class Name

- Name of the class that implements the ScriptService SPI. The SAE instantiates this name when it starts the script service.
- Value—Name of the class
- Default—No value

File/URL

- Shows the content of the script file to be used with the script service.

  To add a script, see *Writing Scripts for Script Services* on page 72.
- Default—No value

Removing a File or URL from a Script Service

To remove a file or URL from a script service:

- In the Script pane, click **Clear**.

  The File/URL field is blank.

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.

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 checks 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.

Adding a Mutex Group

To add a mutex group:

1. In the SDX Admin navigation pane, highlight **Services**, and right-click.

2. Select **New > Mutex** Group.

   The New Mutex Group dialog box appears.

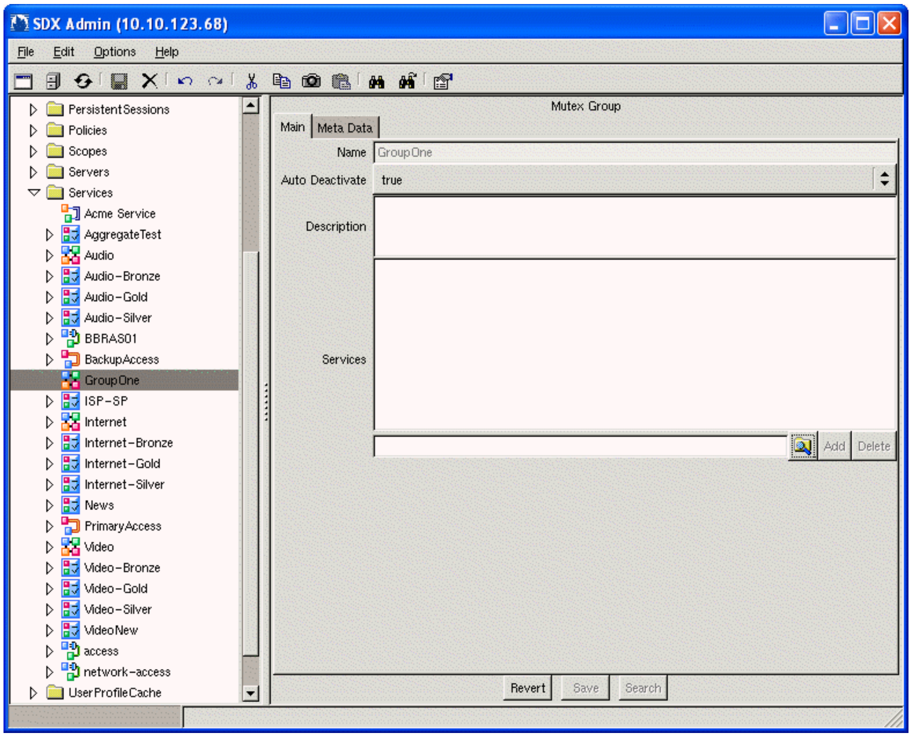
3. Enter a name for the mutex group, and click **OK**.

4. Select either **true** or **false** for Auto Deactivate.

   - true—For any one subscriber, the SAE deactivates a service in the group before activating another service in the group.

   - false—SAE refuses access to a requested service if the subscriber is already using another service in this group.

   An object for the new mutex group appears in the navigation pane, and basic details for the new mutex group appear in the Main tab of the Mutex Group pane.



Mutex Group

Name: GroupOne

Auto Deactivate: true

5. Use the field descriptions in *Mutex Group Fields* on page 81 to configure the mutex group, and then click **Save**.

Mutex Group Fields

Use the fields in this section to configure mutex groups.

Auto Deactivate

- Method that the SAE uses to manage activation of services defined in this group.
- Value
  - true—For any one subscriber, the SAE deactivates a service in the group before activating another service in the group.
  - false—SAE refuses access to a requested service if the subscriber is already using another service in this group.
- Default—No value

Description

- Provides information about this mutex group; keywords that the find utility uses.
- Value—Text
- Default—No value

Services

- Lists the services that the mutex group contains.

  For information about adding services to mutex groups, see *Adding Services to a Mutex Group* on page 81 .

Adding Services to a Mutex Group

You can add multiple services to a mutex group.

NOTE: You must define the service before you can add it to a mutex group. For information about defining services, see *Adding Services* on page 34.

To add a service:

1. Click the magnifying glass below the Services field in the Main tab of the Mutex Group pane.

   The Select Object window appears.

2. Select the services.

   You can shift-click or control-click services to select multiple options.

3. Click **OK**.

   The services appear in the Mutex Group pane.

4. Click **Add**.

   The services appear in the Services field of the Mutex Group pane.

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 *Chapter 14, Defining and Acquiring Values for Parameters*. 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 VRs (see *Configuring Service Scopes* on page 83) and to some types of subscribers (see *SRC-PE Subscribers and Subscriptions Guide, Chapter 13, Configuring Subscribers and Subscriptions with SDX Admin*).

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. Selects the parameters that are defined by only one service scope.
2. If the same parameter is defined by more than one service scope, 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 7, in which three scopes define several parameters for the same service.

Table 7: 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.

Configuring Service Scopes

The tasks to configure a service scope are:

1. Adding Service Scopes on page 84
2. Assigning Services to Service Scopes on page 85
3. Adding Mutex Groups to Service Scopes on page 85
4. Assigning Service Scopes on page 86

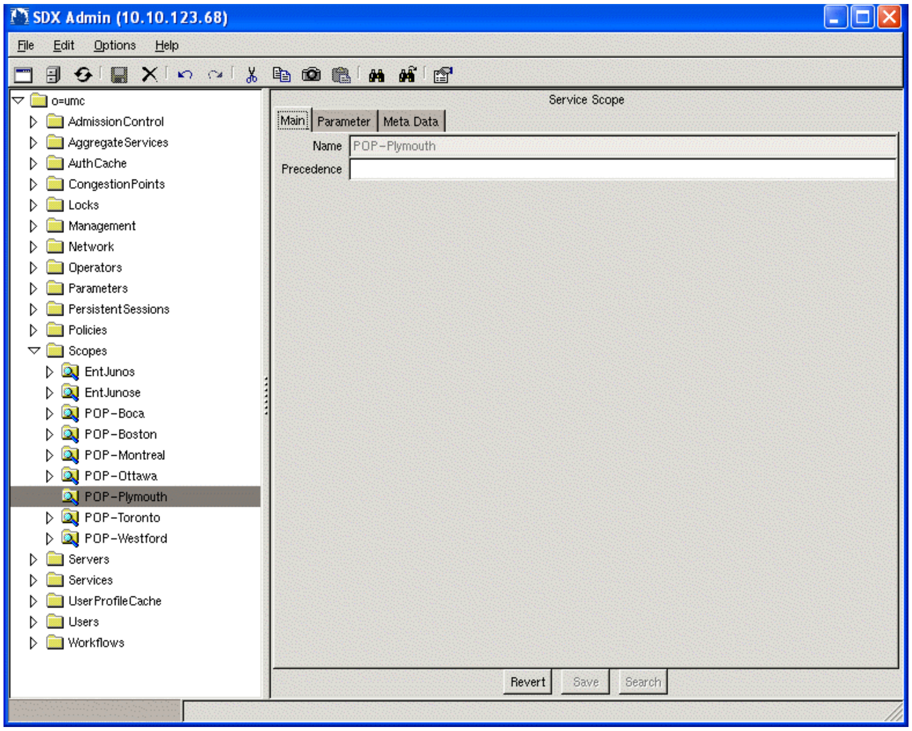
Adding Service Scopes

To add a service scope:

1. In the SDX Admin navigation, highlight **Scopes**, and right-click.
2. Select **New** > **Service Scope**.

   The New Service Scope dialog box appears.
3. Enter a name for the service scope, and click **OK**.

   An object for the new service scope appears in the navigation pane, and basic details for the new service scope appear in the Main tab of the Service Scope pane.



Name POP-Plymouth

Precedence

4. Use the field descriptions in *Service Scope Field* on page 85 to configure the service scope, and then click **Save**.
5. (Optional) You can configure parameters for service scopes. See *Configuring Substitutions* on page 56.

Service Scope Field

Use the field in this section to configure a service scope.

Precedence

- Ranking of this service scope.
- Value—A positive integer; the lower the precedence value, the higher the ranking of the service scope
- Default—No value

Assigning Services to Service Scopes

To assign services to a scope:

1. In the SDX Admin navigation pane, highlight the scope to which you want to assign a service, and right-click.
2. Select **New > SSP Service**.

   The New SSP Service dialog box appears.
3. Select an existing service, or define a new service:
   - Select an existing service from the Service name menu, and click **OK**.
   - Enter a new service name to define a service that appears only in this scope, and click **OK**.

   An object for the assigned service appears subordinate to the service scope in the navigation pane, and details for the new service scope appear in the Main tab of the Service Scope pane.
4. If you defined a new service that appears only in this scope, configure the service, and click **Save** in the pane.

Adding Mutex Groups to Service Scopes

You can add mutex groups to a service scope. If the SAE selects a particular scope, the SAE uses mutex groups in that scope to determine which services it can concurrently activate for a subscriber.

To add a mutex group to a service scope.

1. In the SDX Admin navigation pane, highlight the scope to which you want to assign a service, and right-click.
2. Follow the instructions in *Adding a Mutex Group* on page 79.

Assigning Service Scopes

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

NOTE: You must define the service scope before you can assign it to other objects.

To assign a service scope:

1. In the SDX Admin navigation pane, click the object to which you want to assign the service scope.
2. Click the magnifying glass below the Scope field in the Main tab of the associated pane.

   The Select Object window appears.
3. Select the service scopes.

   You can shift-click or control-click service scopes to select multiple options.
4. Click **OK**.

   The service scopes appear in the associated pane.
5. Click **Add**.

   The service scopes appear in the Services field of the associated pane.

Service Scope Configuration Examples

The following sections provide two practical examples for using scopes to customize your service configuration.

Example: Delivering 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 *o = umc, o = Scopes, l = EntJunos* in the sample data.
2. Add SSP services to the scope, such as those in the sample data under *o = umc, o = Scopes, l = EntJunos*.
3. Assign the scope to one or more enterprises. For example, see *o = umc, o = Users, ou = local, enterpriseName = ABCInc*, and *o = umc, o = Users, ou = local, enterpriseName = Acme.*

If you use an enterprise service portal to manage these organizations, 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 and ABC Inc. from Enterprise Manager Portal, see *SRC-PE Subscribers and Subscriptions Guide, Chapter 24, Managing Services with Enterprise Manager Portal*.

Example: Customizing Generic Services to Particular Regions

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

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

1. Add the Audio-Bronze service, and configure all relevant parameters except the Service IP and Service IP Mask fields in the Parameter tab of the SSP Service pane.

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

2. Set up a scope called scope_boston that specifies the IP address and mask used by VR boston in the Substitution field of the Parameter tab of the Service Scope pane.

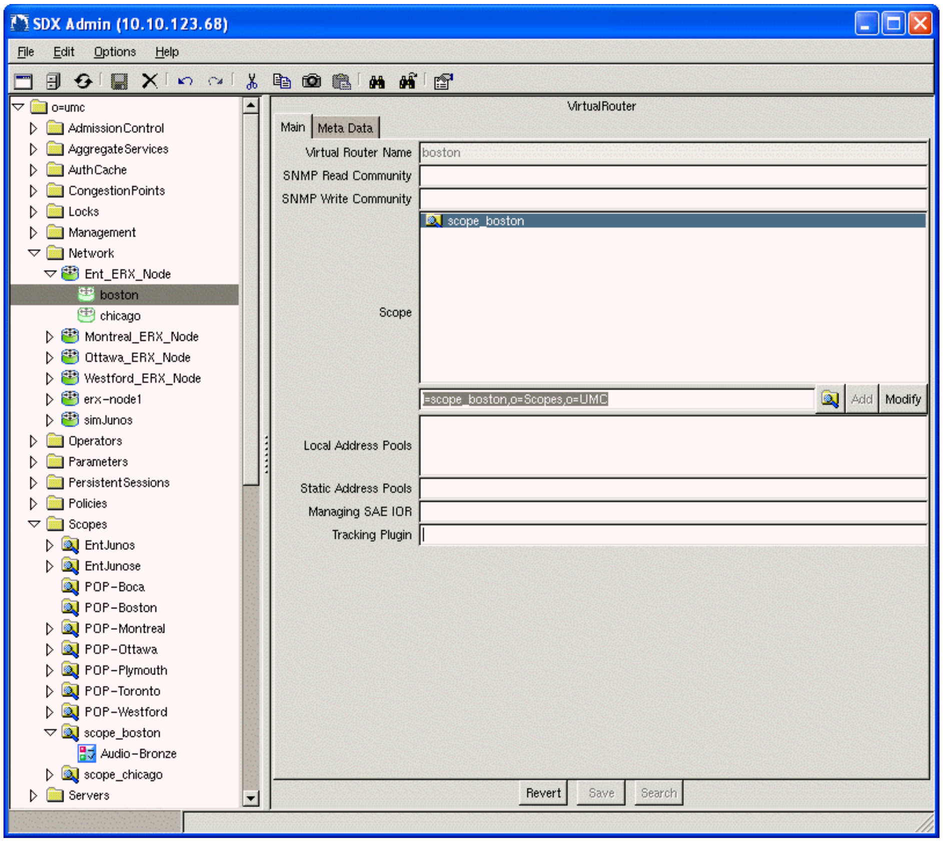
3. Set up a scope called scope_chicago that specifies the IP address and mask used by VR chicago in the Substitution field of the Parameter tab of the Service Scope pane.

4. Assign the service Audio-Bronze to service scopes scope_boston and scope_chicago.

5. Assign the service scope scope_boston to VR boston and the service scope scope_chicago to VR chicago.

Figure 5 shows how this configuration would appear in the SDX Admin navigation pane. 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.

Figure 5: Scopes Configuration Example



Virtual Router Name: boston

SNMP Read Community:

SNMP Write Community:

Scope: scope_boston

=scope_boston,o=Scopes,o=UMC

Local Address Pools:

Static Address Pools:

Managing SAE IOR:

Tracking Plugin:

Allowing 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.

Configuring Permanent Services

To configure a permanent service:

1. In the SDX Admin navigation pane, select the service, and right-click.

   The SSP Service pane appears.

2. In the Permanent field, select **true** from the menu.

3. If you do not want subscribers to deactivate this service, enter the word INVISIBLE in the Category field.

Reviewing Service Status

To use SDX Admin to review the status of a service:

1. In the navigation pane, select a service.
2. In the Main tab, review the value of the Status field.

   A service may have a status of active or inactive:

   - Active—Service accepts new subscriptions.
   - Inactive—Service does not accept new subscriptions.

Restricting Service Activation

You can configure services that the SAE can only activate. 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.

You must complete three actions in this case:

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.

Modifying Services

For information about modifying objects, see *SRC-PE Getting Started Guide, Chapter 43, Using SDX Admin*. For information about configuring a service, see the section that describes how to add that type of service.

Deleting Services

For information about deleting services, see:

- Deleting Services from SDX Admin on page 90
- Deleting Services with Tools Other Than SDX Admin on page 90
- Deleting Services from Scopes on page 91

NOTE: When a value-added service is removed, it is also removed from any mutex group that specifies the service. For information about mutex groups, see *Restricting Simultaneous Activation of Services* on page 78.

Deleting Services from SDX Admin

For information about deleting entries with SDX Admin, see *SRC-PE Getting Started Guide, Chapter 43, Using SDX Admin*.

When you attempt to delete a service, SDX Admin issues a warning message if subscribers have active subscriptions to the service. If you have not configured a workflow for the service and you choose to delete the service regardless of these subscriptions, the SAE deactivates all active subscriptions to that service, and SDX Admin deletes the service and the subscriptions. If you have configured a workflow for a service and a subscriber has an active subscription to the service, you must use the appropriate transactions specified by the workflow utility before you can delete the service. For information about workflow, see the *SRC Application Library Guide*.

Deleting Services with Tools Other Than SDX Admin

To permanently delete a service with an LDAP client other than SDX Admin:

1. Remove the service from the directory with the LDAP client.
2. For each SAE that connects to this directory, update the services in the directory.

   a. Start SAE Web Admin.

      The Home window appears.

   b. Click **Configuration**.

      The Configuration window appears.

   c. Click **Reload Services**.

To make a service unavailable to SRC components such as the SAE, but to leave the service in the directory, set its deleted LDAP attribute to true.

Deleting Services from Scopes

If a scope specifies a service that is defined in *o = Services*, you can delete the service regardless of whether subscribers have active subscriptions to the service. However, if a scope specifies a service that is not defined in *o = Services*, SDX Admin issues a warning message if subscribers have active subscriptions to the service. If you then choose to delete the service regardless of these subscriptions, the SAE deactivates all active subscriptions to that service, and SDX Admin deletes the service from the scope.

Chapter 3

Managing Service Schedules

This chapter provides an overview of service schedules. Topics include:

- Overview of Service Schedules on page 93
- Planning Service Schedules on page 98

You can use the following to create and manage service schedules:

- To use the SRC CLI, see *Chapter 4, Scheduling Services with the SRC CLI*.
- To use the C-Web interface, see *SRC-PE C-Web Interface Configuration Guide, Chapter 24, Scheduling Services with the C-Web Interface*
- To use SRC configuration applications to configure the SRC software on a Solaris platform. See *Chapter 5, Scheduling Services on a Solaris Platform*.

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.

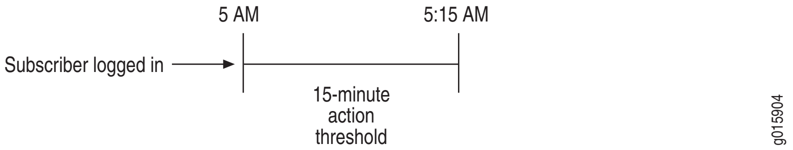
You can also configure an interval after a scheduled time that an action can occur for individual schedules. See *Effective Period for Service Activation or Deactivation* on page 96.

You can configure event-based schedules. See *Adding a Service Schedule with the CLI* on page 104 or *Adding a Service Schedule on a Solaris Platform* on page 116.

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 6.

Figure 6: Sample Action Threshold

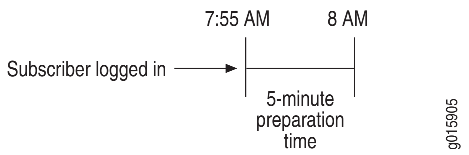


Preparation Time

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 7.

Figure 7: 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.

For information about configuring an authorization plug-in for a service, see *Authorizing Scheduled Services with the CLI* on page 103 or *Authorizing Scheduled Services on a Solaris Platform* on page 115.

For information about configuring authorization schedules, see *Adding a Service Schedule with the CLI* on page 104 or *Adding a Service Schedule on a Solaris Platform* on page 116.

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 *SRC-PE Subscribers and Subscriptions Guide, Chapter 24, Managing Services with Enterprise Manager Portal*.

- 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 in the SRC software distribution in the folder *SDK/doc/idl* or on the Juniper Networks Web site at

  http://www.juniper.net/techpubs/software/management/sdx/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*).

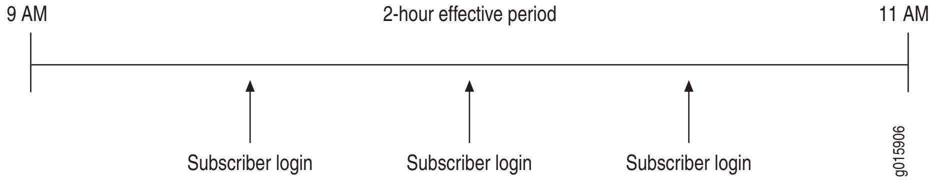
An effective period applies to subscribers who:

- 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 8.

Figure 8: 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.

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 8 shows which subscribers are affected by a schedule.

Table 8: 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.

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.

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.

Planning Schedules

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
   - A defined service scope
   - The subscriber tree

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.

Chapter 4

Scheduling Services with the SRC CLI

This chapter describes how to create and manage schedules for services with the SRC CLI.

You can also use SRC configuration applications to configure the SRC software on a Solaris platform. See *Chapter 5, Scheduling Services on a Solaris Platform*.

Topics in this chapter include:

Setting the Action Threshold and Preparation Time with the 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
   ```

Authorizing Scheduled Services with the 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
   ```

Adding a Service Schedule with the 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* on page 105.

   - Specify the actions.

     See *Setting the Action* on page 108.

Setting the Time Schedule

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 guidelines in *Guidelines for Entering Time Values* on page 107.

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
   ```

Guidelines for Entering Time Values

When you enter time schedules, you can use the values in the following list. See *Setting the Time Schedule* on page 105 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.

Setting the Action

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;
    substitution [substitution...]; }
```

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 with the CLI* on page 103.

   ```
   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 in the SRC software distribution in the folder *SDK/doc/idl* or on the Juniper Networks Web site at

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

   For more information about substitutions and schedules, see *Example: Configuring Different Service Tiers for Different Days with the CLI* on page 109.

   For information about the syntax for substitutions, see *Chapter 14, Defining and Acquiring Values for Parameters*.

Defining Attributes for Service Activation

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.

   For more information about subscription attributes, see the *Subscription.html* file in the SAE core portal API documentation in the *SDK/doc/sae/net/juniper/smgt/sae/portal* directory in the SRC software distribution or on the Juniper Networks Web site at

   http://www.juniper.net/techpubs/software/management/sdx/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
   ```

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 1of 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
    ```

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
   ```

Chapter 5

Scheduling Services on a Solaris Platform

This chapter describes how to manage service schedules for your SRC configuration with the SRC configuration applications that run only on Solaris platforms. You can also use the SRC CLI that runs on Solaris platforms and the C-series Controller to configure service schedules. See *Chapter 4, Scheduling Services with the SRC CLI*. For information about service schedules, see *Chapter 3, Managing Service Schedules*.

Topics in this chapter include:

Authorizing Scheduled Services on a Solaris Platform

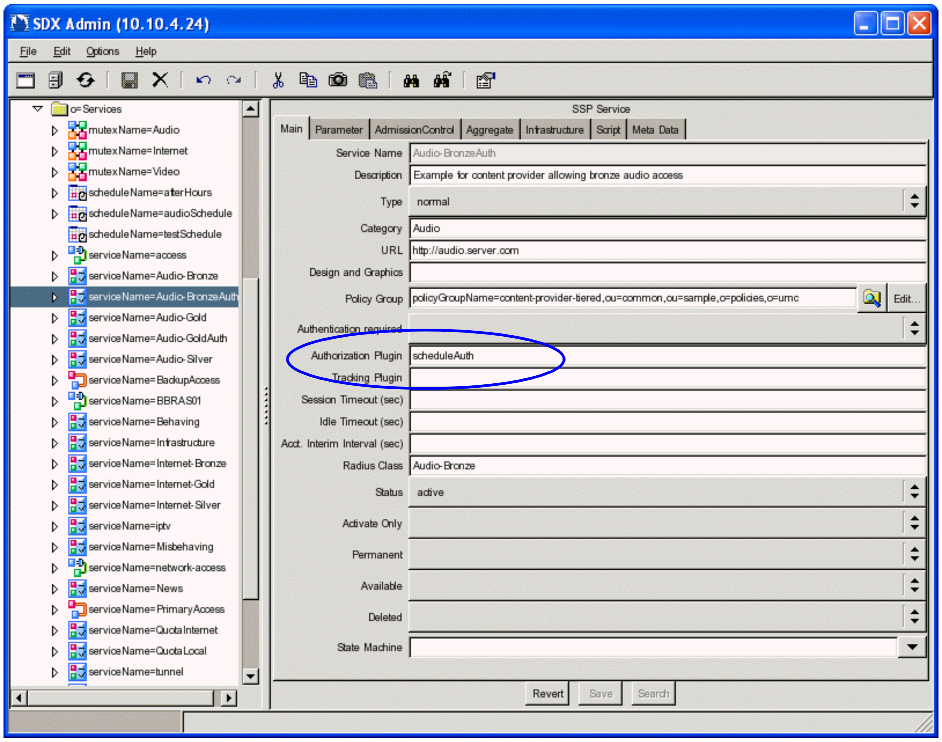
You can configure an authorization plug-in to authorize a scheduled service by specifying the name of the plug-in that authorizes the schedule in the service definition. The default schedule authorization plug-in is named scheduleAuth.

To use SDX Admin to define an authorization plug-in for a service:

1. In the navigation pane, select the service.
2. In the Main tab of the content pane, in the Authorization Plugin field enter the name of the authorization plug-in that will authorize the schedule for this service.

3. Click **Save**.

In the following configuration example, an Internet-Bronze service uses the scheduleAuth authorization plug-in.



Service Name: Audio-BronzeAuth
Description: Example for content provider allowing bronze audio access
Type: normal
Category: Audio
URL: http://audio.server.com
Design and Graphics:
Policy Group: policyGroupName=content-provider-tiered,ou=common,ou=sample,o=policies,o=umc
Authentication required:
Authorization Plugin: scheduleAuth
Tracking Plugin:
Session Timeout (sec):
Idle Timeout (sec):
Acct. Interim Interval (sec):
Radius Class: Audio-Bronze
Status: active
Activate Only:
Permanent:
Available:
Deleted:
State Machine:

Adding a Service Schedule on a Solaris Platform

Table 9 lists the objects for which you can create a service schedule in SDX Admin.

Table 9: Objects for Service Schedules

Type of Directory Object	Distinguished Name
Scopes	*o = umc, o = Scopes, l = < Scope >*
Services	*o = umc, o = Services*
Retailers	*o = umc, o = Users, l = < Retailer >*
Enterprises	*o = umc, o = Users, l = < Retailer > , l = < Subscriber Folder >*
Users in an Enterprise	*o = umc, o = Users, l = < Retailer > , l = < Subscriber Folder > , l = < User >*

To use SDX Admin to add a service schedule:

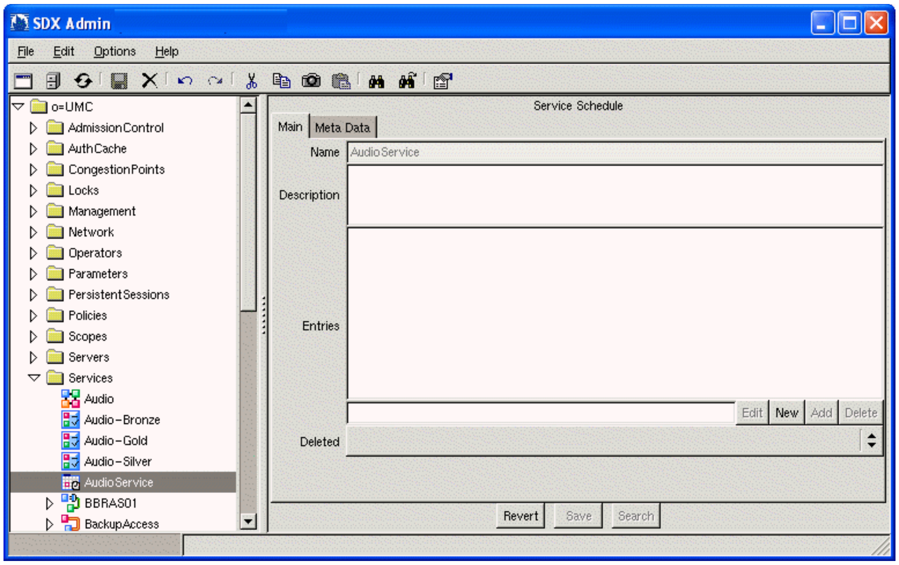
1. In the navigation pane, right-click one of the objects listed in Table 9, and select **New > Service Schedule**.

   The New Service Schedule dialog box appears.

2. Enter a unique name for the service schedule, and click **OK**.

   An object for the new service schedule appears in the navigation pane, and basic details for the service schedule appear in the Main tab of the Service Schedule pane.

   In the Main tab of the Service Schedule pane, you can edit a service schedule by adding, editing, and modifying entries to the service schedule.



Name AudioService

3. Edit or accept default values for the service schedule fields.

   See *Service Schedule Fields* on page 117.

4. Click **Save**.

Service Schedule Fields

In SDX Admin, you can modify the following fields in the content pane for a service schedule.

Description

- Describes the service schedule.
- Value—Text
- Default—No value

Entries

- Lists each scheduled entry.

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

  To add an entry, see *Creating an Entry for a Schedule on a Solaris Platform* on page 118.

Deleted

- Specifies the availability of this entry to other SRC components connected to the directory.
- Value
  - Blank—Other SRC components can access this entry in the directory.
  - True—Other SRC components cannot use this entry in the directory, although the object still exists.
  - False—Other SRC components can access this entry in the directory.
- Default—Blank

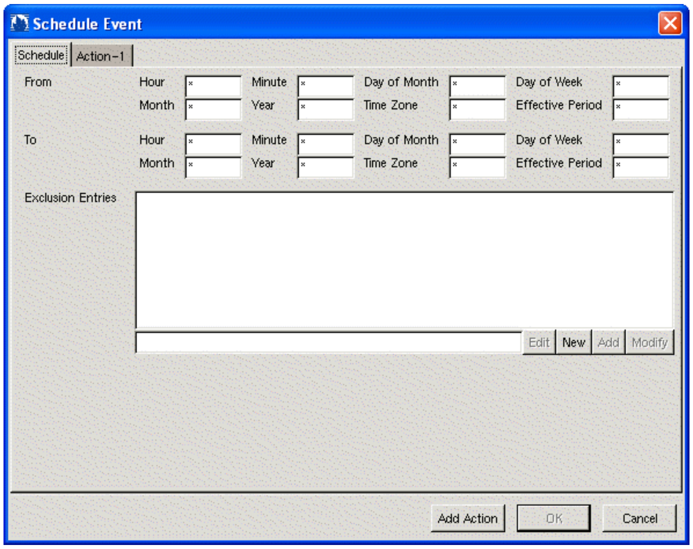
Creating an Entry for a Schedule on a Solaris Platform

A number of schedule entries, or rules, comprise each service schedule.

To use SDX Admin to create an entry:

1. In the Service Schedule pane, click **New**.

   The Schedule Event dialog box appears.



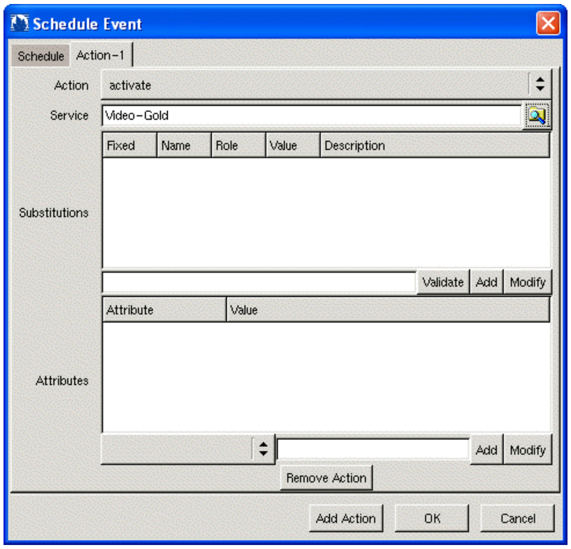
2. In the Schedule tab of the Schedule Event dialog box:

   - Edit the From date and time information.
   - (Optional for services that have an authorization plug-in configured) Edit the To date and time information.
   - (Optional) Specify time exclusions.

     See *Setting the Time Schedule* on page 120.

3. In the Action tab of the Schedule Event dialog box, fill in the fields.

   See *Setting the Action* on page 124.



Action: activate

Service: Video–Gold

4. To add another action to the entry, in the Action tab click **Add Action**.
5. To remove an action from the entry, in the Action tab, click **Remove Action**.
6. After you complete entries in the Schedule Event dialog box, click **OK** to return to the Service Schedule pane.
7. To add the entry to the list of entries in the service schedule, click **Add**.
8. After you enter all schedules and actions, click **OK** in the Schedule Event dialog box.

   The completed schedule appears in the Service Schedule pane.

Setting the Time Schedule

When you set up a time schedule, you specify:

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

Sample Time Definitions

Table 10 provides descriptions of common schedule configurations.

Table 10: Sample Schedules

Type of Event to Schedule	Sample Configuration Description
One-time event	Specify all time parameters except day of the week. Specify the time zone if it is different from the time zone for the SAE.
Recurring time-of-day event, such as a service that is activated every morning at 8:00 AM	Specify only the hour and minute parameters. All other parameters should have an *. Specify the time zone if it is different from the time zone for the SAE.
Working-hours service	Define two schedules: one to activate the service and one to deactivate the service. In each schedule, specify the hour and minute, and specify the day of week as mon-fri. All other parameters should have an *. Specify the time zone if it is different from the time zone for the SAE. **NOTE:** Subscribers must be logged in at the time that the service is activated.
No service restrictions on holidays	Define schedules for working-hour services as described above, and add exceptions for specified holidays, such as the first of January.

Configuring the Time Schedule

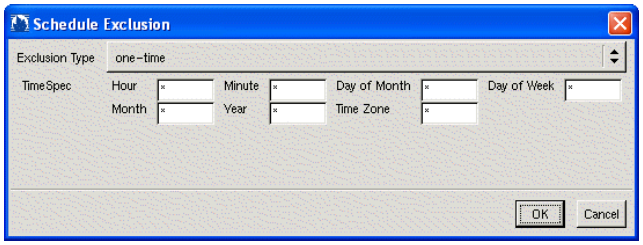
To configure the time schedule:

1. Enter information in the fields in the Schedule tab in the Schedule Event dialog box.

   Use the guidelines in *Guidelines for Entering Time Values* on page 121 and the field descriptions in *Time Values* on page 122.

2. To specify an exclusion from the schedule, in the Schedule tab of the Schedule Event dialog box, click **New** under Exclusion Entries.

   The Schedule Exclusion dialog box appears.



Exclusion Type: one–time

TimeSpec: Hour * Minute * Day of Month * Day of Week * Month * Year * Time Zone *

3. Enter information in the fields in the Schedule Exclusion dialog box; then click **OK**.

   Use the guidelines in *Guidelines for Entering Time Values* on page 121 and the field descriptions in *Time Values* on page 122,

Guidelines for Entering Time Values

When you enter information in the Schedule Event and Schedule Exclusion dialog boxes, you can use the values in the following list. See *Time Values* on page 122 for a description of the fields.

NOTE: Dates in the To section of the dialog box 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 section are ignored.

- *—Asterisks are interpreted as follows:
  - Minutes and hours:
    - 0 if used in the From or To fields of a scheduled event
    - First or last if used in the Time Spec field of a schedule exclusion
  - Time zones—Local SAE time zone
  - All other fields—First through last
  - For fields in the To section of the dialog box, * for the end time is equivalent to “deny service activation after this start date.”
  - For dates in the From section of the dialog box, * is equivalent to “deny service activation before this end date.”
- Range of numbers or letters separated by a hyphen. The range is inclusive; for example, 1-5 for the hour specifies hours 1, 2, 3, 4, and 5. A range of mon-wed specifies Monday, Tuesday, and Wednesday.
- List of numbers, letters, or ranges separated by commas. For example, 1,2,5,9 or 0-4,8-12 or mon-wed,fri-sat.

- 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 field 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.

Time Values

In SDX Admin, you can modify the following fields in the Schedule Event and Schedule Exclusion dialog boxes. For information about general guidelines that apply to these fields, see *Guidelines for Entering Time Values* on page 121.

Exclusion Type

- Interval to exclude from the schedule specified in the Schedule Event dialog box.
- Value
  - one-time—Exclusion for a single time; for example, for a holiday
  - period—Exclusion for a time range; for example, a number of days
- Guidelines—This field applies only to the Schedule Exclusion dialog box. Effective periods do not apply to schedules for excluded times.
- Default—one-time

Hour

- Hour of the day in the indicated month in which to schedule the event or exclusion.
- Value—0–23
- Default—*

Minute

- Minutes past the indicated hour in which to schedule the event or exclusion.
- Value—0–59
- Default—*

Day of Month

- Day of the month in which to schedule the event or exclusion.
- Value—1–31
- Default—*

Day of Week

- Day of the week in which to schedule the event or exclusion.
- Value
  - 0–6, with 0 representing Sunday and each subsequent number representing the next day of the week.
  - First three letters of the name of the day
- Default—*
- Examples—For Saturday and Sunday, specify one of the following:
  - sat, sun
  - 6, 0

Month

- Month of the year in which to schedule the event or exclusion.
- Value
  - 1–12
  - First three letters of the name of the month
- Default—*
- Examples—For January, specify one of the following:
  - jan
  - 1

Year

- Year in which to schedule the event or exclusion.
- Value—Four integers that indicate the year
- Default—*

Effective Period

- 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.
- Value—Number of minutes in the range 0–153722867280912
- Guidelines—The effective period applies only to schedules configured for an object under *o = Users*.

  The effective period does not apply to schedules for excluded times; the entry is not present in the Schedule Exclusion dialog box.
- Default—*

Time Zone

- Name of the time zone to use in the schedule.
- Value
  - * —Local time zone of the SAE.
  - An offset to GMT in the format:

    GMT (+|-) (hh:mm | hh mm | hh)

    hh—<hour>

    mm—<minute>
- Default—Time zone for the SAE
- Examples
  - Canada/Eastern or America/New York
  - GMT +5—Sets the time zone to 5 hours ahead of GMT

Setting the Action

In SDX Admin, you can modify the following fields in the Action tab of the Schedule Event dialog box.

Action

- Type of action.
- Value
  - activate—Service is activated at the time specified in the entry schedule.
  - deactivate—Service is deactivated at the time specified in the entry schedule.
  - deny—New activation requests for this service during the specified entry period are denied; current sessions are not affected.
  - deny&deactivate—New activation requests for this service during the specified period are denied; in addition, current sessions are deactivated when the specified time occurs.
- Guidelines—The deny and the deny&deactivate values apply only to services that have an authorization plug-in configured.
- Default—No value

Service

- Name of the service.
- Value—Text

  You can type the name of the service or click the folder icon to display the Select Service dialog box. Select the service in the dialog box, and click OK.
- Default—No value

Substitutions

- Substitutions to be used when the service is activated. Substitutions apply only to service activations.

  For more information, see the activateService method of the SAE external interface in the SAE CORBA remote API documentation in the SRC software distribution in the folder *SDK/doc/idl* or on the Juniper Networks Web site at

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

  For more information about substitutions and schedules, see *Changing or Removing the Name of a Service Associated with a Schedule* on page 125.
- Value—An entry in valid substitution format.

  For information about the syntax for substitutions see *Chapter 14, Defining and Acquiring Values for Parameters*.
- Default—No value

Attributes

- Defines attributes that are set before the service is activated.

  Subscription attributes apply only to service activations.

  For more information about subscription attributes, see the *Subscription.html* file in the SAE core portal API documentation in the *SDK/doc/sae/net/juniper/smgt/sae/portal* directory in the SRC software distribution or on the Juniper Networks Web site at

  http://www.juniper.net/techpubs/software/management/sdx/api-index.html
- Value—Use the Up or Down arrow below the Attributes box to select an attribute, and then click **Add** to add the attribute to the action.
  - sessionName—Name of the service session.
  - sessionTag—Tag that can be used for accounting purposes.
  - sessionTimeout—Session timeout to be used when the service is activated. The service session is deactivated when this timeout expires.
  - downStreamBandwidth—Attribute used by SRC Admission Control Plug-In (SRC-ACP) to specify the rate of traffic between the network and the subscriber.
  - upStreamBandwidth—Attribute used by SRC-ACP to specify the rate of traffic between the subscriber and the network.

Changing or Removing the Name of a Service Associated with a Schedule

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.

Deleting a Schedule Entry

To delete a schedule entry:

- In the Service Schedule pane, right-click an item in the Entries box, and click **Delete**.

Deleting a Schedule Exclusion Entry

To delete an exclusion entry:

- In the Schedule Event dialog box, right-click an exclusion under Exclusion Entries, and select **Delete**.

Editing a Schedule Entry

To edit a schedule entry:

1. In the Service Schedule pane, click an item in the Entries box.

   The entry appears in the box below the Entries field.

2. Click **Edit** to display the Schedule Event dialog box for the entry.

3. Fill in the Schedule Event dialog box. See *Creating an Entry for a Schedule on a Solaris Platform* on page 118.

4. Click **Modify** to replace the selected entry with the entry that you have just edited.

Editing a Schedule Exclusion Entry

To edit an exclusion entry:

1. In the Schedule Event dialog box, highlight an exclusion under Exclusion Entries, and click **Edit**.

2. Make changes in the Exclusion Schedule dialog box, and click **OK**.

Example: Configuring Different Service Tiers for Different Days

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.

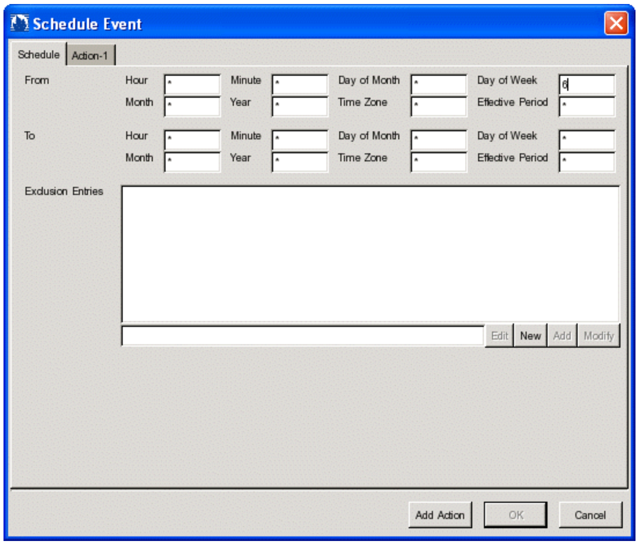
For demonstration purposes, the sample schedule is configured under *o = Services* to make the service schedule available to all subscribers to the two audio services. It is assumed that subscribers are continuously logged in to the system to access the audio services.

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. In the SDX Admin navigation pane, right-click *o = Services* and select **New > Service Schedule**.
2. In the New Service Schedule dialog box, enter a name for the schedule; for example, audioSchedule.

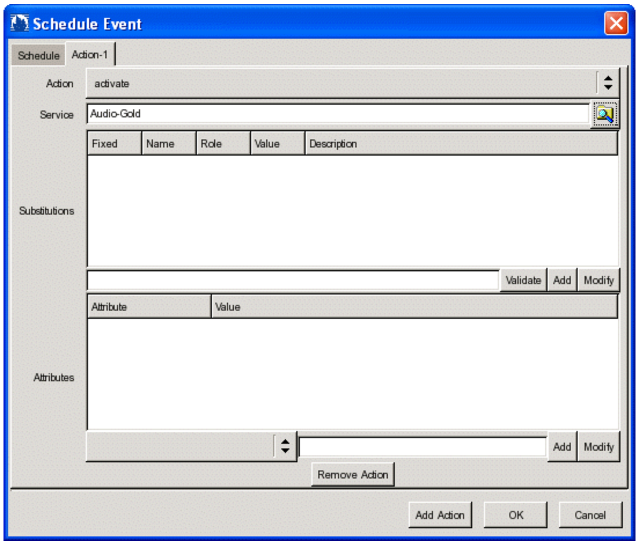
   The name of the service appears in the Service Schedule pane.
3. In the Description field of the Service Schedule pane, enter a description of the schedule.
4. In the Service Schedule pane, click **New**.

   The Schedule Event dialog box appears.
5. In the Schedule tab, specify the day of the week as 6 for Saturday.



From	Hour	*	Minute	*	Day of Month	*	Day of Week	6
	Month	*	Year	*	Time Zone	*	Effective Period	*
To	Hour	*	Minute	*	Day of Month	*	Day of Week	*
	Month	*	Year	*	Time Zone	*	Effective Period	*

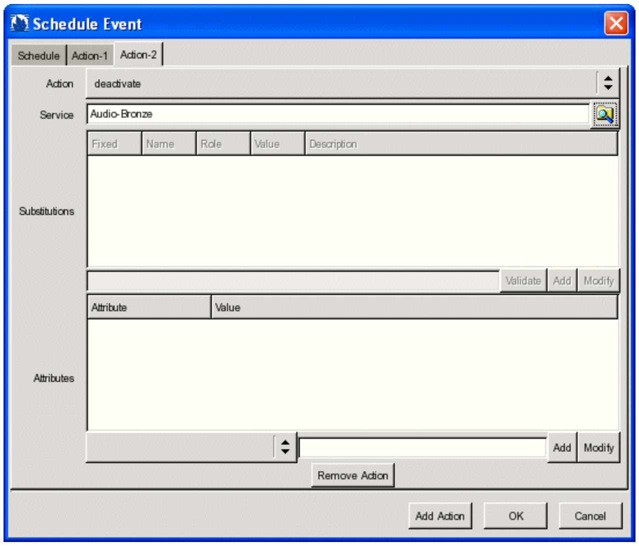
6. Click the **Action-1** tab, and specify **activate** for the Audio-Gold service.



Action: activate

Service: Audio-Gold

7. Click **Add Action**, and then click the **Action-2** tab. Specify **deny and deactivate** for the Audio-Bronze service.



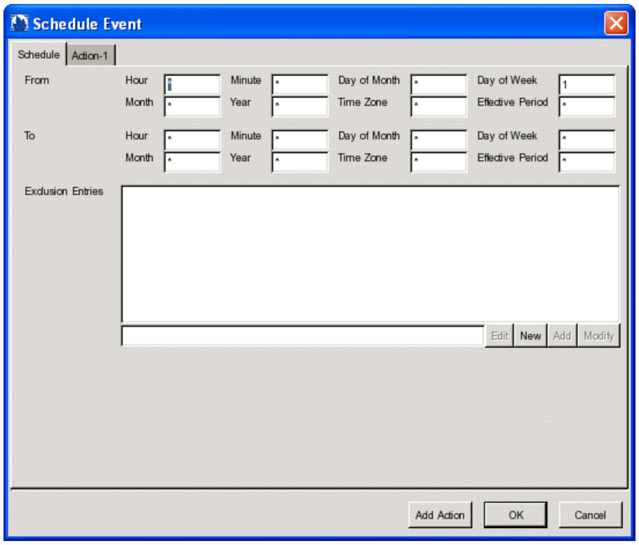
Action: deactivate

Service: Audio-Bronze

8. Click **OK**, and then in the Service Schedule pane click **Add** to add the schedule entry.

9. In the Service Schedule pane, click **New** to add another schedule entry.

10. In the Schedule tab of the Schedule Event dialog box, specify the day of the week as 1 for Monday.



Schedule Event

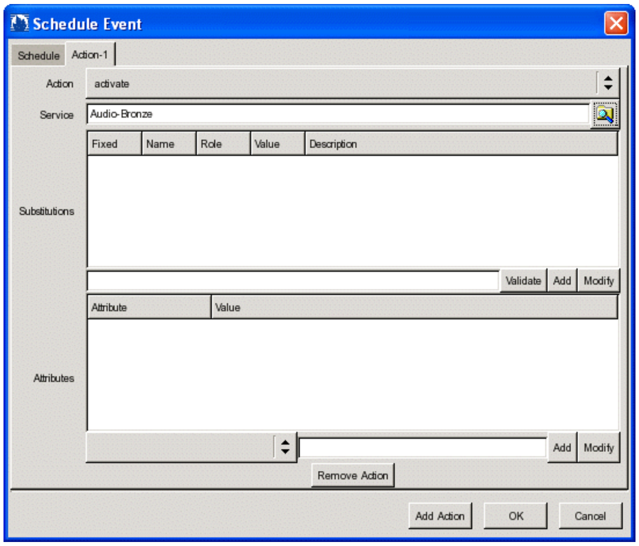
Schedule | Action-1

From: Hour | Minute * | Day of Month * | Day of Week 1
Month * | Year * | Time Zone * | Effective Period *

To: Hour * | Minute * | Day of Month * | Day of Week *
Month * | Year * | Time Zone * | Effective Period *

Exclusion Entries

11. Click the **Action-1** tab, and specify **activate** for the Audio-Bronze service.



Schedule Event

Schedule | Action-1

Action: activate

Service: Audio-Bronze

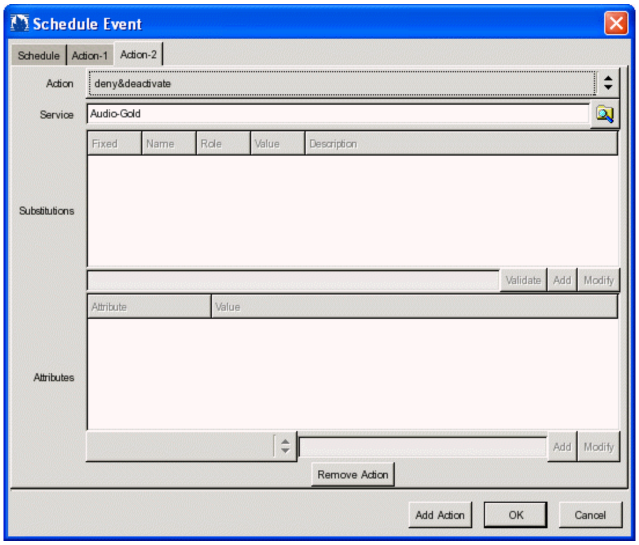
Substitutions

Fixed	Name	Role	Value	Description

Attributes

Attribute	Value

12. Click **Add Action**; and then click the **Action-2** tab. Specify **deny and deactivate** for the Audio-Gold service.

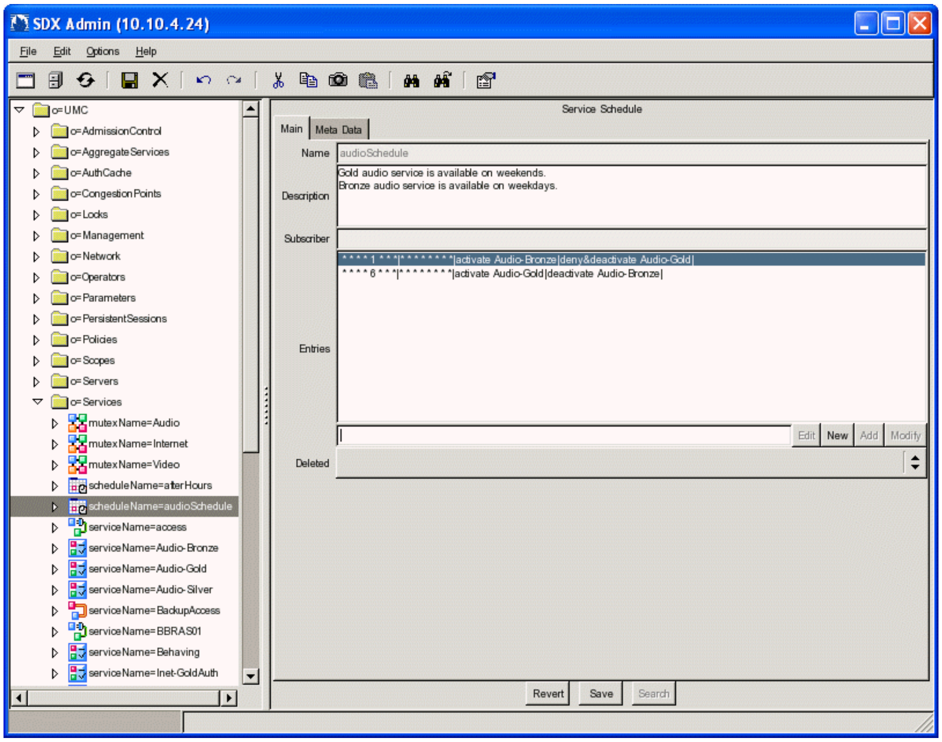


Action deny&deactivate

Service Audio-Gold

13. Click **OK**, and then in the Service Schedule pane click **Add** to add the schedule entry.

    The Service Schedule pane displays the new schedule:



Name: audioSchedule

Description: Gold audio service is available on weekends.
Bronze audio service is available on weekdays.

Subscriber:

Entries:

```
* * * * 1 * * *|* * * * * * * *|activate Audio-Bronze|deny&deactivate Audio-Gold|
* * * * 6 * * *|* * * * * * * *|activate Audio-Gold|deactivate Audio-Bronze|
```

Deleted:

14. Click **Save** to save the schedule.

Example: Configuring a Service to Be Active During Nonwork Hours

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 1of 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.

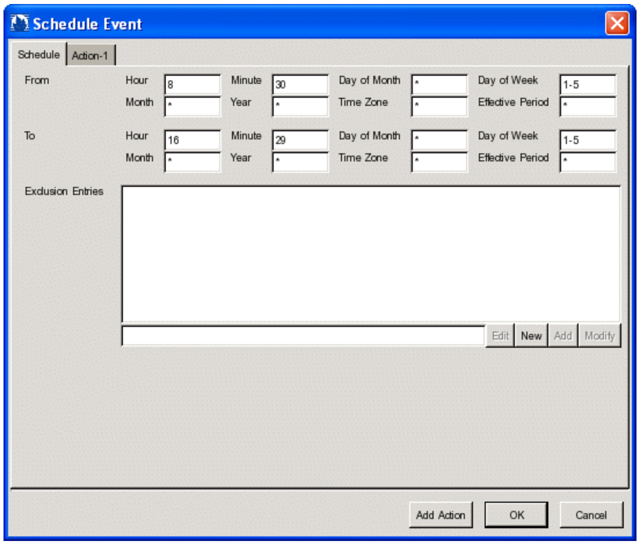
For demonstration purposes, the sample schedule is configured under *o = Services* to make the service schedule available to all subscribers to the Internet-GoldAuth service.

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

1. In the SDX Admin navigation pane, right-click *o = Services* and select **New > Service Schedule**.
2. In the New Service Schedule dialog box, enter a name for the schedule; for example, afterHours.

   The name of the schedule appears in the Service Schedule pane.
3. In the Description field of the Service Schedule pane, enter a description for the schedule.
4. In the Service Schedule pane, click **New**.

   The Schedule Event dialog box appears.
5. In the Schedule tab, specify that the schedule start at 8 AM on Monday through Friday and end at 4:29 PM (that is, 16:29) on Monday through Friday.

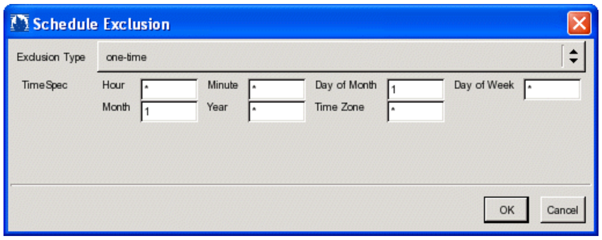


From	Hour	8	Minute	30	Day of Month	*	Day of Week	1-5
	Month	*	Year	*	Time Zone	*	Effective Period	*
To	Hour	16	Minute	29	Day of Month	*	Day of Week	1-5
	Month	*	Year	*	Time Zone	*	Effective Period	*

6. Under Exclusion Entries, click **New**.

   The Schedule Exclusion dialog box appears.

7. In the Schedule Exclusion dialog box, specify a one-time exclusion for January 1, and click **OK**.



Schedule Exclusion

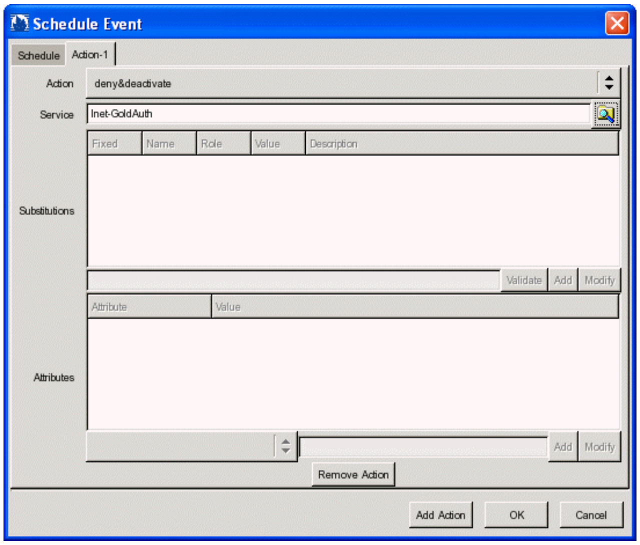
Exclusion Type: one-time

TimeSpec

Hour	*	Minute	*	Day of Month	1	Day of Week	*
Month	1	Year	*	Time Zone	*		

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

8. Click the **Action-1** tab, and specify **deny and deactivate** for the Internet-GoldAuth service.



Schedule Event

Schedule | Action-1

Action: deny&deactivate

Service: Inet-GoldAuth

Substitutions

Fixed	Name	Role	Value	Description

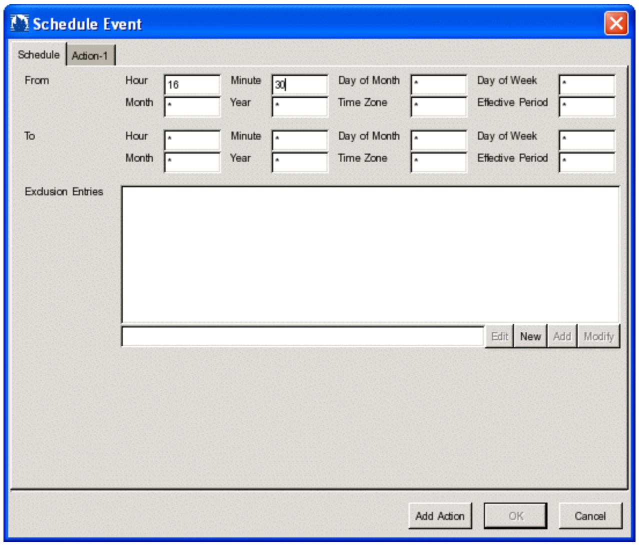
Attributes

Attribute	Value

9. Click **OK**, and then in the Service Schedule pane click **Add** to add the schedule entry.

10. In the Service Schedule pane, click **New** to add another schedule entry.

11. In the Schedule tab of the Schedule Event dialog box, specify 4:30 PM (that is, 16:30).



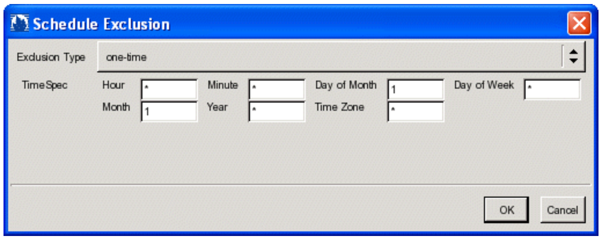
From — Hour: 16; Minute: 30; Day of Month: *; Day of Week: *
Month: *; Year: *; Time Zone: *; Effective Period: *

To — Hour: *; Minute: *; Day of Month: *; Day of Week: *
Month: *; Year: *; Time Zone: *; Effective Period: *

Exclusion Entries

12. Under Exclusion Entries, click **New**.

13. In the Schedule Exclusions dialog box, specify a one-time exclusion for January 1, and click **OK**.

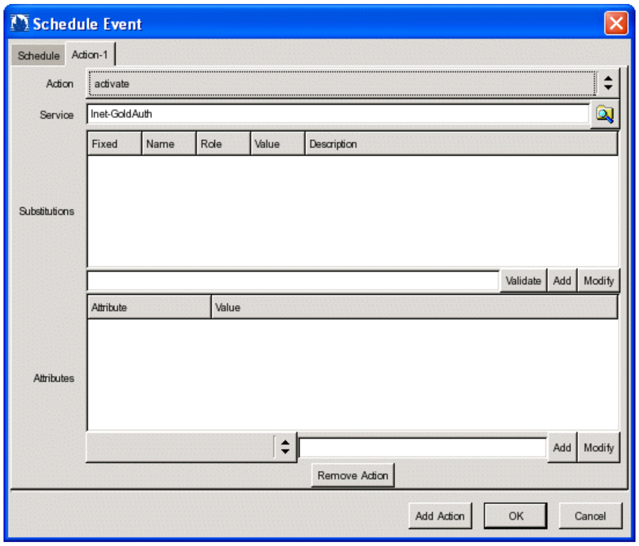


Exclusion Type: one-time

TimeSpec — Hour: *; Minute: *; Day of Month: 1; Day of Week: *
Month: 1; Year: *; Time Zone: *

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

14. Click the **Action-1** tab, and specify **activate** for the Internet-GoldAuth service.

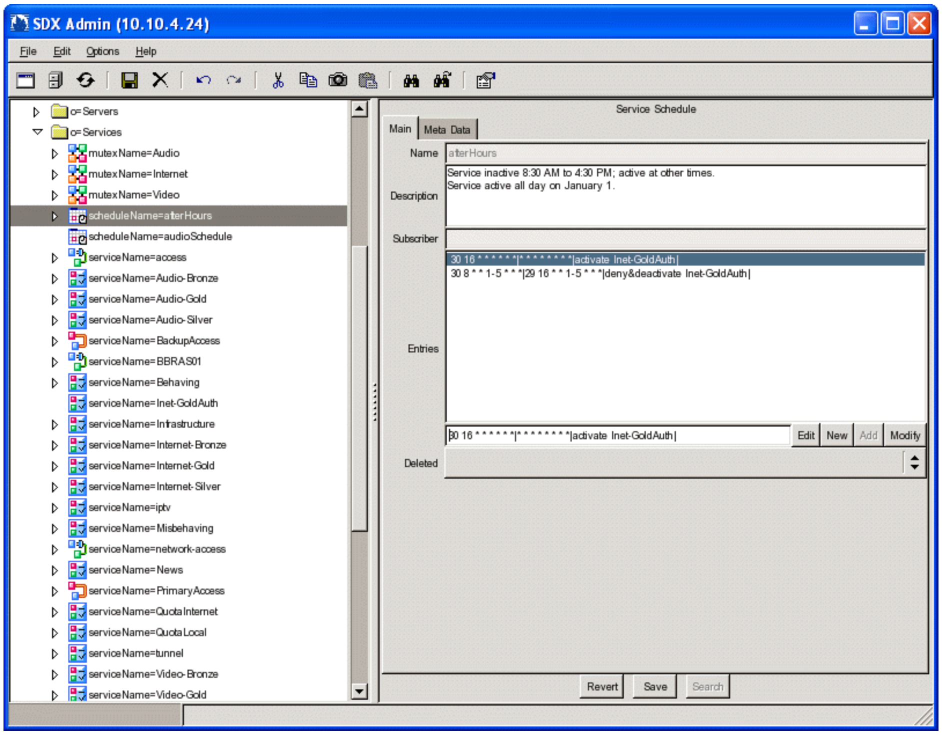


Action: activate

Service: Inet-GoldAuth

15. Click **OK**, and then in the Service Schedule pane click **Add** to add the schedule entry.

    The Service Schedule displays the new schedule:



Name: afterHours

Description: Service inactive 8:30 AM to 4:30 PM; active at other times.
Service active all day on January 1.

Subscriber:

Entries:

```
30 16 * * * * * *|* * * * * * * *|activate Inet-GoldAuth|
30 8 * * 1-5 * * *|29 16 * * 1-5 * * *|deny&deactivate Inet-GoldAuth|
```

```
30 16 * * * * * *|* * * * * * * *|activate Inet-GoldAuth|
```

Deleted:

16. Click **Save** to save the schedule.

Example: Configuring a Service to Be Available for a Specified Interval

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:

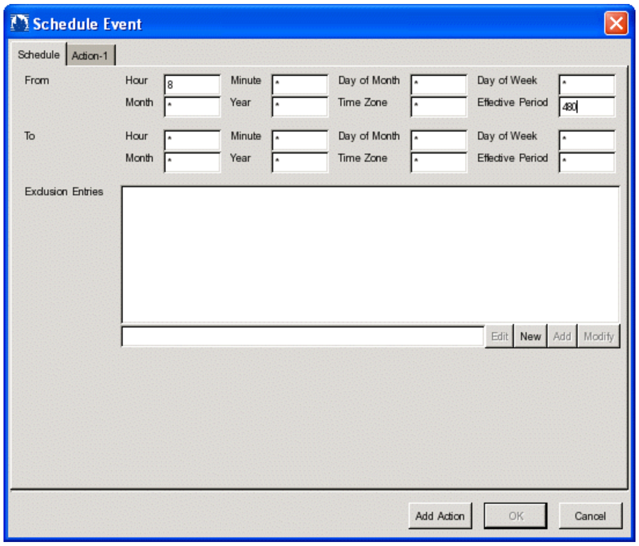
1. Create a schedule by right-clicking *o=Services* in SDX Admin; then select **Service Schedule**.

   The Schedule Event dialog box appears.

2. In the Schedule tab in the Schedule Event dialog box:

   - Specify the time when the service is first available–8.
   - Specify how long the service is to be available–480.

The schedule is to be available from 8 AM to 4 PM; that is, 8 hours, which equals 480 minutes.



Schedule Event

From	Hour	8	Minute	*	Day of Month	*	Day of Week	*
	Month	*	Year	*	Time Zone	*	Effective Period	480
To	Hour	*	Minute	*	Day of Month	*	Day of Week	*
	Month	*	Year	*	Time Zone	*	Effective Period	*

Exclusion Entries

3. In the Action-1 tab:

   a. In the Action field, select **activate**.

   b. In the Service field, select a service.

4. Click **OK** to add the schedule; then in the Service Schedule pane click **Add** to add the service.

5. Click **Save** to save the schedule.

Part 2
Defining Policies to Manage Traffic

Chapter 6

Policy Management Overview

This chapter provides an overview of the policy management feature. Topics include:

Overview of Policy Management

The SRC software works with Juniper Networks routers and PacketCable Multimedia Specification (PCMM)–compliant cable modem termination system (CMTS) platforms to provide differentiated quality of service (QoS). The SRC software uses policies to define how the router or the CMTS device treats subscriber traffic. Policy management is responsible for defining policies and deploying the policies on a router or CMTS device.

Router Policy Features Supported

This section describes the features that the SRC policy management software supports on JUNOS routing platforms and on JUNOSe routers. For information about features supported on CMTS devices, see *Delivering QoS Services in a Cable Environment* on page 157.

JUNOS Routing Platform Features

The SRC software supports the following features on JUNOS routing platforms:

- JUNOS class-of-service (CoS)

  Allows you to provide differentiated services. You can assign forwarding classes to different applications, set a loss priority, define which packets are placed into each output queue, schedule the transmission service level for each queue, and manage congestion using a random early detection (RED) algorithm. You can also configure a shaping rate for interfaces.

  For complete information about how this feature works on the router, see the *JUNOS Network Interfaces and Class of Service Configuration Guide*.

- Firewall filter

  Allows you to control packets transiting the router to a network destination and packets destined for and sent by the router.

  For complete information about how this feature works on the router, see the *JUNOS Policy Framework Configuration Guide*.

- Policing, or rate limiting

  Enables you to limit the amount of traffic that passes into or out of an interface. Policing is designed to thwart denial-of-service (DoS) attacks. It applies two types of rate limits on the traffic:

  - Bandwidth—Number of bits per second permitted, on average.

  - Maximum burst size—Maximum size permitted for bursts of data that exceed the bandwidth limit.

  For complete information about how this feature works on the router, see the *JUNOS Policy Framework Configuration Guide*.

- Adaptive Services PIC (ASP)

  Supports stateful firewall and network address translation (NAT) services:

  - Stateful firewall—Type of firewall filter that considers state information derived from previous communications and other applications when evaluating traffic.

  - NAT—Security procedure for concealing host addresses on a private network behind a pool of public addresses.

  For complete information about how this feature works on the router, see the *JUNOS Services Interfaces Configuration Guide*.

- Port mirroring

  Allows you to control traffic on the router by mirroring traffic with a preconfigured mirroring port and filtering with a specific policy.

  For complete information about how this feature works on the router, see the *JUNOS Policy Framework Configuration Guide*.

JUNOSe Router Features

The SRC software supports the following policy management features on JUNOSe routers:

- Policy routing

  Allows the router to classify a packet on ingress and make a forwarding decision based on that classification, without performing the normal routing table processing.

- Rate limiting

  Provides bandwidth management by enforcing line rates below the physical line rate of the port and setting limits on packet flows.

- QoS classification and marking

  Marks packets in a packet flow so that the QoS application can provide traffic-class queuing.

- Packet forwarding

  Forwards packets in a packet flow.

- Packet filtering

  Drops packets in a packet flow.

For complete information about how these features work on the router, see the *JUNOSe Policy Management Configuration Guide*.

For more information about using the SRC software to manage QoS services on JUNOSe routers, see *SRC-PE Solutions Guide, Chapter 1, Managing Tiered and Premium Services with QoS on JUNOSe Routers with the SRC CLI*.

Default Policies and Service Policies

The policy management feature provides two types of policies that make it possible for you to control when the policies are deployed; this feature provides dynamic deployment of policies. The two types of policies are:

- Default policies—Are attached to a router interface when the SAE begins to manage the interface, before subscribers activate services. Default policies define the subscriber’s initial network access. Typically, they block access to value-added services, restrict a subscriber’s bandwidth, or restrict network access altogether.

  If you are using the captive portal in a PCMM environment, you do not need default policies.

- Service policies—Are attached to an interface when a subscriber activates a service; they take priority over the default policy. Service policies allow access to value-added services or provide higher bandwidth. (When you create a service policy, you assign a lower precedence number to the policy rule so that it is preferred over the default policy.)

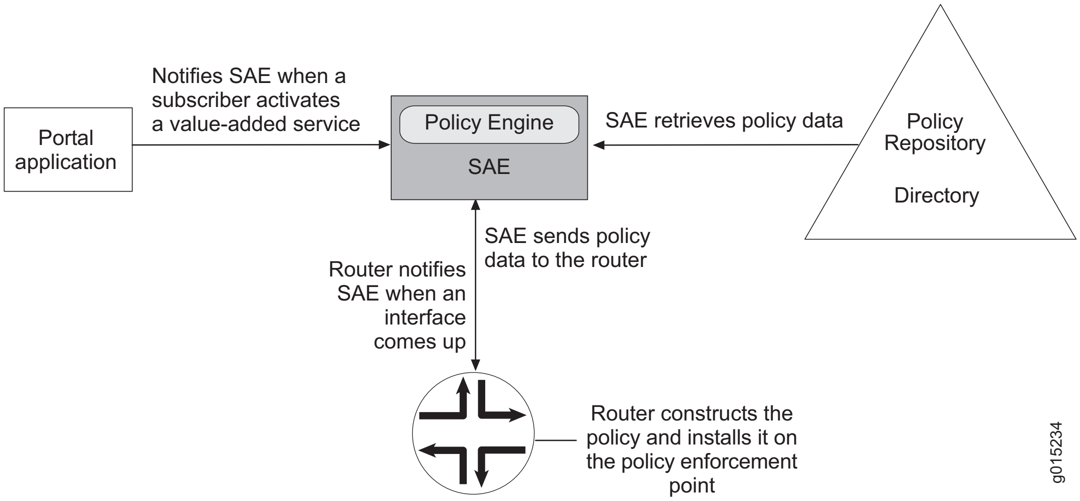
Where you reference the policies determines whether it is a default policy or a service policy. Default policies are referenced in interface classification scripts. Service policies are referenced in value-added service definitions.

How Policies Are Installed on the Router

The policy engine in the SAE makes decisions about the deployment of policies on the router. When the SAE needs to install a policy on the router, it retrieves the policy data from the directory, processes the data, and sends the data to the router. The router uses the data to construct the policy, and then it applies the policy as instructed by the SAE.

Figure 9 gives an overview of how policies are installed on the router.

Figure 9: Installing Policies on the Router


Notifies SAE when a subscriber activates a value-added service
Portal application
Policy Engine
SAE
SAE retrieves policy data
Policy Repository
Directory
SAE sends policy data to the router
Router notifies SAE when an interface comes up
Router constructs the policy and installs it on the policy enforcement point
g015234

Installing Default Policies

When an interface comes up on the router, the SAE runs the interface classification script to determine whether it manages the interface. If the interface is managed—that is, controlled by—the SAE, the SAE sends the default policy referenced in the interface classification script to the router.

Installing and Removing Service Policies

When a subscriber activates a service (for example, video-gold), the portal application notifies the SAE to activate that service. The SAE obtains the policy data associated with the service and sends the data to the router. The router constructs and installs the appropriate policies.

When the subscriber deactivates the service, the portal application passes the request to the SAE, and the SAE notifies the router to remove the policies for the service.

Reloading Default Policies

The SAE reapplies default policies when:

- The definition of a default policy changes.
- The interface classification criteria change.

When the SAE is triggered to reload default policies, it generates default policies for each interface that was previously reported as up. If the default policies have changed compared with the previously applied policies, the current default policies (if any) are removed and the new policies are applied.

NOTE: This behavior means that the SAE also must keep track of unmanaged interfaces to handle changes in the interface classification script.

Policy List Sharing

Policy list sharing is supported on JUNOS routing platforms and on JUNOSe routers that are managed using the COPS-PR router driver. Policy sharing allows the same policy list to be attached to multiple interfaces. Before the SAE modifies policies that are attached to an interface, installs policies on an interface, or removes policies from an interface, it checks whether the requested combination of policy rules already exists on the router.

- If the combination exists, the SAE changes the policy attachment of the interface to use the existing policy. Using an existing policy increases router performance because the router does not have to construct a new policy.

  The router maintains policy counters when it changes policy attachments. To generate accounting data, the SAE reads the policy counters before it deactivates a policy.

- If the combination does not exist, the SAE sends the policy data to the router. The router either creates a new list (if the interface is not managed yet) or modifies the policy list currently attached to the interface.

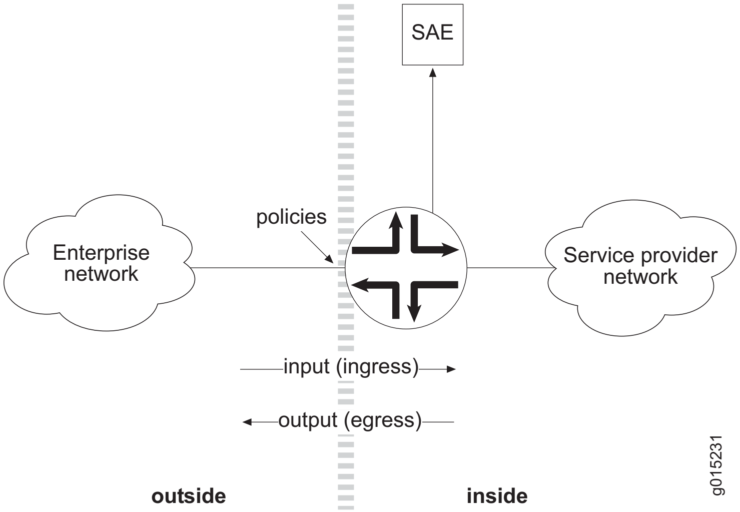
If a policy list is no longer referenced by a session, the SAE removes it from the interface.

Network Perspective for Creating Policies

When you create a policy, you indicate where the policy is applied on the router. You can apply policies to the ingress (input) side of the interface, to the egress (output) side of the interface, to both the ingress and egress sides of the interface, or, in the case of JUNOS scheduler policy rules, you can attach the policy to the interface without indicating direction. Typically, policies are applied to subscriber-facing interfaces.

Figure 10 shows a sample network diagram with an enterprise network and a service provider network. 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.

Figure 10: Network Perspective for Creating Policies



Collecting Accounting Statistics

You can specify whether accounting data is collected for the actions specified in a policy rule. If you specify that accounting data is collected, the SAE begins collecting accounting information when a service that uses the policy rule is activated. When the service is deactivated, the SAE sends the accounting records to the RADIUS accounting server or to a plug-in.

When you specify multiple actions for accounting, the SAE adds the accounting data for individual actions together to obtain a summary accounting record for that interface direction.

Accounting is not available for all actions. For example, the NAT action does not provide accounting.

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:

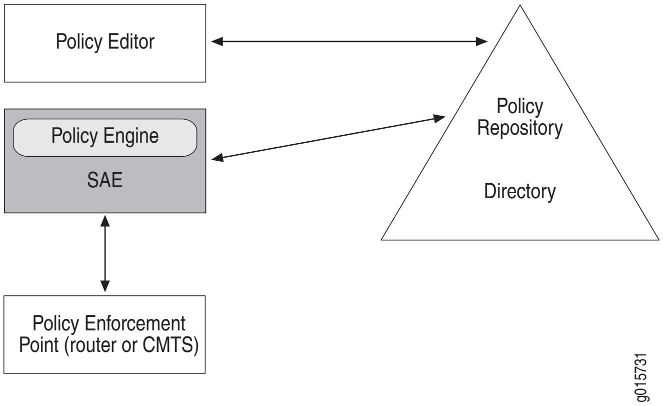
- Policy Editor—Defines and deploys policies
- Policy engine—Resides on the SAE and makes policy decisions (policy decision point)

- 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 11 shows the components of the policy management system. As shown:

1. Policy Editor 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 11: Policy Management Components



Policy Editor

Policy Editor is one of the applications that you use to define policies. It dynamically changes the panes that it provides to you, based on your input. It can show or hide policy object attributes as you interact with it. For example, when you choose the TCP or UDP protocols, the source and destination ports are shown; otherwise, they are not shown.

Policy Editor also allows you to store policy data in a directory server or in files. By storing data in files, you can create a backup of the repository or transfer policies from one repository to another.

See *Chapter 7, Using Policy Editor*.

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.

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 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.

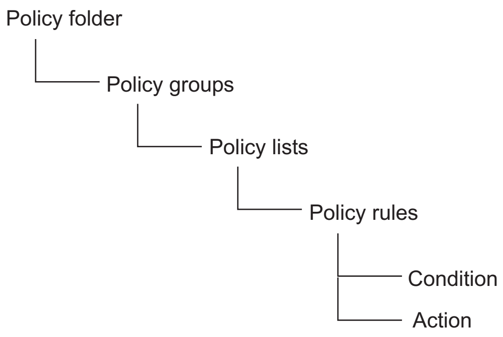
The SRC policy model also lets you simplify policy configuration for policy conditions that classify traffic. For JUNOSe and PCMM policies, you can combine different conditions that classify traffic and configure these conditions to use a single action. In addition for JUNOSe policies, you can create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

For more information about multiple classifiers and expanded classifiers, see *Policy Conditions* on page 152.

Policy Objects

The SRC policy model is made up of objects that are organized as shown in Figure 12.

Figure 12: 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, 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.
- 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 and PCMM devices support one type of policy rule. 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 13 shows the types of conditions and actions that are available for JUNOS policy rules. Figure 14 shows the types of conditions and actions that are available for JUNOSe policy rules. The color and user packet class actions are available for JUNOSe policy rules on IPv6 routers. Figure 15 shows the types of conditions and actions that are available for PCMM policy rules.

Figure 13: JUNOS Policy Rules with Supported Conditions and Actions

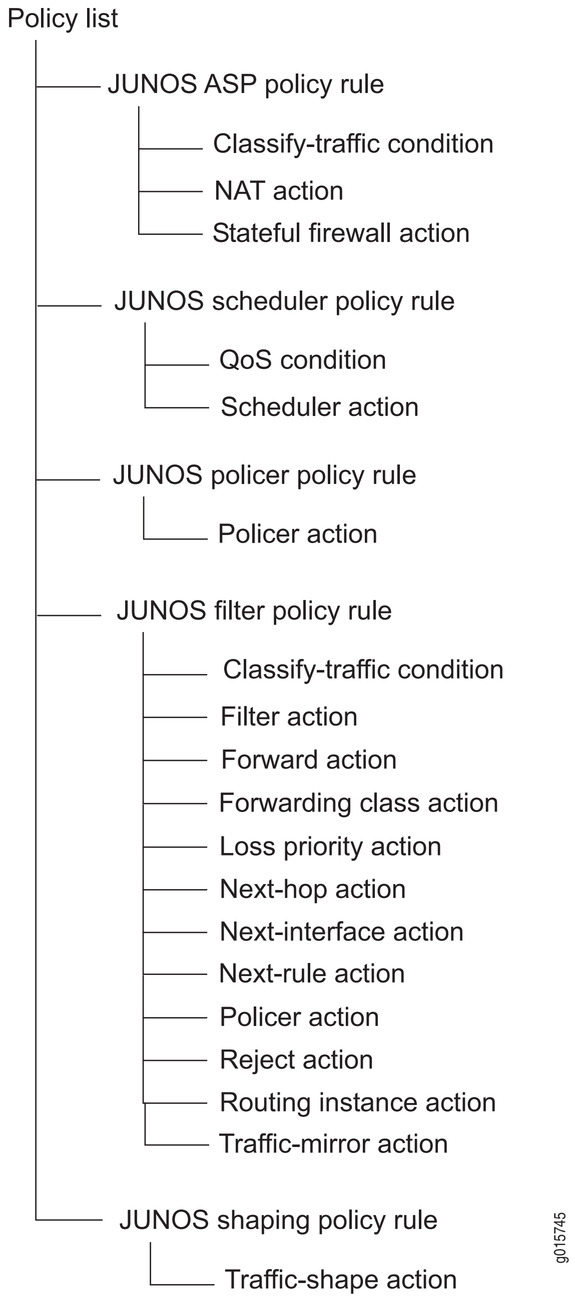


Figure 14: JUNOSe Policy Rules with Supported Conditions and Actions

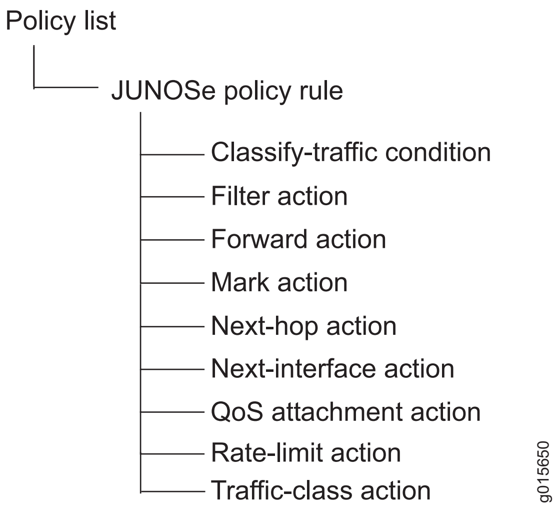
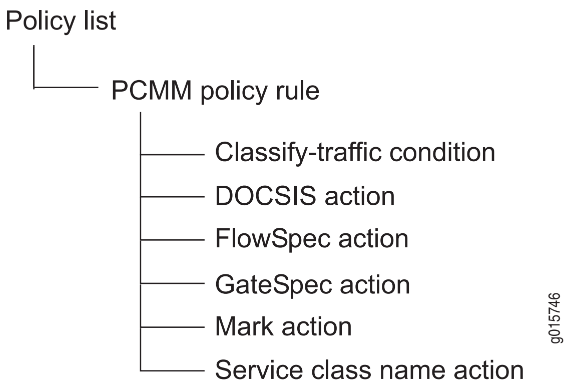


Figure 15: PCMM Policy Rules with Supported Conditions and Actions



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 160.

Multiple Classifiers

JUNOSe 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

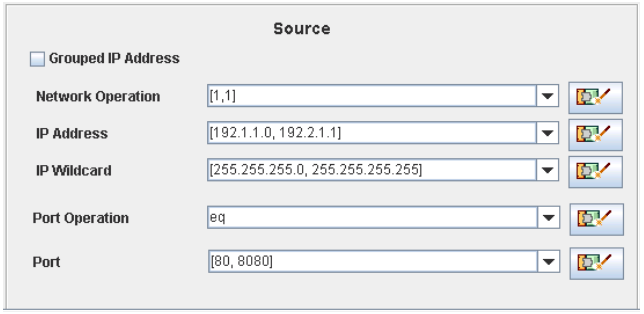
For JUNOSe policies, you can create classify-traffic conditions that the SAE expands into multiple classifiers before it installs the policy on the router. If you enter a comma-separated list of values in the source and destination network (IP address, mask, and IP operation) or port fields (for port-related protocols), the software creates a classifier for each possible combination of address and port. Note that 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 JUNOSe policies.

For example, the source configuration in the classify-traffic condition in Figure 16 would cause the condition to be expanded into four classifiers that have the following combination of source addresses and source ports:

```
192.1.1.0/255.255.255.0 eq 80
192.1.1.0/255.255.255.0 eq 8080
192.2.1.1/255.255.255.0 eq 80
192.2.1.1/255.255.255.0 eq 8080
```

Figure 16: Classify-Traffic Condition Example for Expanded Classifiers



Source

☐ Grouped IP Address

Network Operation	[1,1]
IP Address	[192.1.1.0, 192.2.1.1]
IP Wildcard	[255.255.255.0, 255.255.255.255]
Port Operation	eq
Port	[80, 8080]

Policy Actions

JUNOS policy rules and PCMM policy rules can have multiple actions. JUNOSe 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 150. The following table is a description of all actions.

Table 11: Policy Actions

Action	Type of Rule	Description
Color	JUNOSe	Specifies the color attribute that is applied to the packet when it passes through the router that is running IPv6.
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.
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.

Table 11: Policy Actions (continued)

Action	Type of Rule	Description
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.
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 that is running IPv6.

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 JUNOSe policy rule, you can set up a policy list to take two corresponding actions on a packet. To do so, you create a JUNOSe 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.

Policy LDAP Schema Model

The policy information model is based on the Policy Core Information Model (PCIM) that is mapped to the Policy Framework LDAP core schema by the IETF. SRC software extends this model in such a way to be very close to the policy model used by the router. A policy folder might be the base of the policy subtree (*o=policies, o=umc*) or an organizationalUnit object, underneath the policy base. Such a policy folder contains group objects consisting of one or many policy lists that contain one or many policy rules. A policy rule consists of policy actions and policy conditions.

The objects policy group, policy list, and policy rule are mapped to structural object classes. Each of those classes is derived from the object class policy. This abstract policy object class is inherited from dlm1ManagedElement, which is the top class of the CIM. The policy actions and policy conditions are mapped to auxiliary classes that are attached to the object policyRule. The classes policyActionAuxClass and policyConditionAuxClass are the top classes for any policy action and policy condition. SSP service objects point through the DN pointer to one policy group.

For detailed information about the SRC LDAP schema, see the documentation in the SRC software distribution in the folder */SDK/doc/ldap* or on the Juniper Networks Web site at

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

Delivering QoS Services in a Cable Environment

This section 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 12 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 12: 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
Non-real-time polling service (NRTPS)	For upstream service flows. 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.	Standard Internet traffic that requires high throughput, and traffic that requires variable-sized data grants on a regular basis, such as high-bandwidth FTP.	Traffic priority Request transmission policy Maximum sustained traffic rate Maximum traffic burst Minimum reserved traffic rate Assumed minimum reserved-traffic-rate packet size Nominal polling interval
Real-time polling service (RTPS)	For upstream service flows. Analogous to NRTPS, except that the fixed polling interval is typically very short. Offers request opportunities that meet the service flows' real-time needs and allows the cable modem to specify the size of the desired grant.	Real-time traffic that generates variable-sized data packets on a periodic basis and has inflexible latency and throughput requirements. Applications include Moving Pictures Experts Group (MPEG) video.	Request transmission policy Maximum sustained traffic rate Maximum traffic burst Minimum reserved traffic rate Assumed minimum reserved-traffic-rate packet size Nominal polling interval Tolerated poll jitter

Table 12: DOCSIS Service Flow Scheduling Types (continued)

Type	Description	Suitable Traffic Type(s)	QoS Parameters
Unsolicited grant service (UGS)	For upstream service flows. 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.	Real-time traffic that generates fixed-size data packets on a periodic basis. Applications include voice over IP (VoIP)	Request transmission policy Unsolicited grant size Grants per interval Nominal grant interval Tolerated grant jitter
Unsolicited grant service with activity detection (UGS-AD)	For upstream service flows. A hybrid of the UGS and RTPS scheduling types. ■ 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	Request transmission policy Nominal polling interval Tolerated poll jitter Unsolicited grant size Grants per interval Nominal grant interval Tolerated grant jitter
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

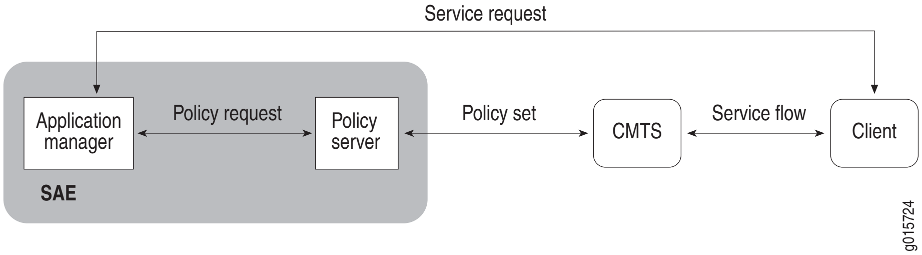
The PCMM specification defines three types of clients, and defines a client as a logical entity that can send or receive data. The SRC software supports client type 1, which represents endpoints such as PC applications or gaming consoles that lack specific QoS awareness or signaling capabilities. Client 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 17: 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. In Policy Editor, 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* on page 218 in *Chapter 10, Configuring and Managing Policies with the SRC CLI*.
- *Specifying the PCMM Classifier Type* on page 296 in *Chapter 11, Configuring and Managing Policies with Policy Editor*.
- *SRC-PE C-Web Interface Configuration Guide, Chapter 21, Configuring and Managing Policies with the C-Web Interface, Specifying the PCMM Classifier Type.*

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.

Using Policy Editor, 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 all of the service flow scheduling types and QoS parameters described in Table 12 on page 157. See one of the following:

- *Configuring DOCSIS Actions* on page 247 in *Chapter 10, Configuring and Managing Policies with the SRC CLI*.
- *Configuring DOCSIS Actions* on page 153 in *SRC-PE C-Web Interface Configuration Guide, Chapter 21, Configuring and Managing Policies with the C-Web Interface*
- *Configuring Color Actions* on page 320 in *Chapter 11, Configuring and Managing Policies with Policy Editor*.

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* on page 274 in *Chapter 10, Configuring and Managing Policies with the SRC CLI*.
- *Configuring Service Class Name Actions* on page 165 in *SRC-PE C-Web Interface Configuration Guide, Chapter 21, Configuring and Managing Policies with the C-Web Interface*.
- *Configuring Service Class Name Actions* on page 358 in *Chapter 11, Configuring and Managing Policies with Policy Editor*.

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 13 shows how the FlowSpec service types map to the DOCSIS service scheduling types.

Table 13: 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 14 shows the parameters that you can set for each service type.

Table 14: 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.

Chapter 7
Using Policy Editor

This chapter describes how to use Policy Editor. Topics include:

Overview of Policy Editor

Policy Editor allows you to define policies in the policy repository. It is a Java swing application that acts as an X11 application on Solaris. As a result, it complies with and provides complete X11 functionality.

Key Mapping

Figure 18 shows the keyboard modifier map and key map table that Policy Editor uses to convert event code into key symbols (keysyms) in X11.

Figure 18: Keyboard Modifier Map

```
$ xmodmap
xmodmap:  up to 2 keys per modifier, (keycodes in parentheses):

shift       Shift_L (0x31),  Shift_R (0x3d)
lock        Caps_Lock (0x41)
control     Control_L (0x24),  Control_R (0x5e)
mod1        Alt_L (0x3f),  Alt_R (0x60)
mod2
mod3        Num_Lock (0x61)
mod4
mod5
```

You generate this map by entering the following command in UNIX:

```
xmodmap
```

To customize your keyboard, the key mapping can be changed with standard X11 utilities.

Nonroot Users

Nonroot operators can access Policy Editor. The home directory for nonroot operators stores operator customization and log files.

The files and directories in *pom_install_dir* are not changed when various operators use Policy Editor. Nonroot operators who want to use Policy Editor must have read/write permission on files and directories in *pom_install_dir.* The Policy Editor default installation gives read/write permission to all users (that is, user, group, and others). The operating system commands are used to restrict access as required.

When Policy Editor is started, the directory *POM_USER_DIR*, which is *<$HOME>/.UMC/pom,* is used. The <$HOME> variable is the user's home directory. The *<$HOME>* directory must exist. The *.UMC/pom* directory path is created if it does not exist.

There are two subdirectories in *POM_USER_DIR*:

- *etc*—Contains the customization and configuration for this user
- *var*—Contains the log directory and file for this user

Exception Handling for the Directory

If Policy Editor detects a problem, such as a connection loss, when it accesses the directory, it displays a dialog box that describes the LDAP problem. If such a problem occurs, you can back up modified data to a different directory or to a file, or you can ignore the changes and open a different data source. You must either open or cancel the current operation.

The open operation opens the Open Directory Server dialog box, allowing you to change the connection parameters if it is necessary to connect to another directory.

Typically, there is one primary directory in the main office of the service provider and various secondary directories in regional centers. The service provider synchronizes policies between the master directory and the slave directories. As a result, changes to policies in each of the directories are controlled and propagated to the other required directories.

Providing Data Security

Policy Editor implements data security through:

- Simple authentication
- Access control

Simple authentication occurs when Policy Editor sends the fully qualified distinguished name (FQDN) of the client and the client's clear-text password to the directory.

The directory supports access control that defines and monitors different clients' access privileges on LDAP entries.

Working with Policy Data Files

Policy Editor does not support the reading of policy data files produced with earlier versions of the SDX software. This feature may work under some circumstances; for example, when Policy Editor is used in JUNOSe mode.

Multi-User and Multi-Instance Concurrence

Multiple operators with multiple instances can use Policy Editor simultaneously. Concurrence control manages different instances or different operators while they perform operations on the same policy object at the same time.

Concurrence control is transparent; that is, the system does not notify you of simultaneous changes to the same object. However, be aware that, if multiple operators work on the same objects at the same time, the object properties that one operator configures can be overwritten by another operator; that is, the system stores the second operation in the directory.

Table 15 shows various concurrence control scenarios. In this table both operation 1 and operation 2 are performed on the same object by two different operators or application instances. The commitment of operation 2 is later than that of operation 1.

Table 15: Concurrence Control Scenarios

Operation 1	Operation 2	Changes in the Directory
Delete	Delete	The object is deleted from the directory. Operation 2 is ignored.
Add	Add	The object created by operation 2 is stored in the directory.
Modify	Modify	The modification made by operation 2 is stored in the directory.
Delete	Modify	The modification made by operation 2 is stored in the directory. The object is re-created.
Modify	Delete	The object is deleted from the directory.

For the scenario of delete/delete, the object is deleted from the directory at the commitment of operation 1. When the same request is received from operation 2, it is silently ignored. For all other operations, operation 2 (that is, the latest operation) always prevails.

The directory ensures consistency for each entry but not across multiple entries. See *SRC-PE Integration Guide, Chapter 2, Overview of LDAP Integration*.

In summary, the directory does not maintain consistency among objects. To maintain consistency, all operators must be in contact with each other and agree on what the final state of the data should be.

Starting Policy Editor

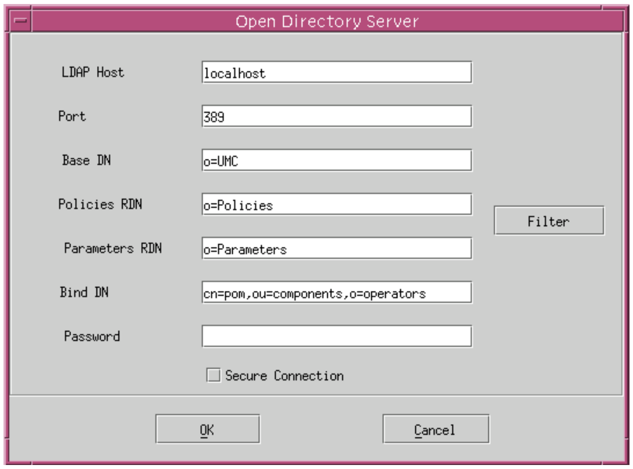
To start Policy Editor, at the UNIX prompt (#), type:

```
cd /opt/UMC/pom/bin
./pomgui
```

To open a policy repository:

1. Click **File**, select **Open**, and click **Directory Server**.

   The Open Directory Server dialog box appears.



LDAP Host: localhost
Port: 389
Base DN: o=UMC
Policies RDN: o=Policies
Parameters RDN: o=Parameters
Bind DN: cn=pom,ou=components,o=operators
Password:
□ Secure Connection

2. Fill in the fields. See Table 16.

Table 16: Open Directory Server (LDAP Connection) Fields

Field	Description	Notes
LDAP Host	IP address or hostname of the directory	You can connect to only one directory at a time.
Port	Port number for the directory host connection	Default—389 Range—1–65535 Type—Integer
Base DN	Distinguished name of the base policy information in the directory	Default—*o = umc* If the base DN does not exist, you are prompted to create it. To create a base DN, the parent object must exist.
Policies RDN	Relative distinguished name of the policies information in the directory	This name is relative to the base DN.
Parameters RDN	Relative distinguished name of the base global parameters information in the directory	Default—*o = Parameters* This name is relative to the base DN.
Bind DN	Distinguished name used for binding to the directory	Default—*cn = pom, ou = components, o = operators, o = umc*
Password	Password associated with the bind DN	Default—pom

You can also filter the information (click Filter in the Open Directory Server dialog box). This filter function is the same as that of the filter function available through the Tools menu. (See *Filtering Searches* on page 177.)

You can also select Secure Connection if the directory host you connect to supports Transport Layer Security (TLS). Selecting Secure Connection forces encryption of the directory connection.

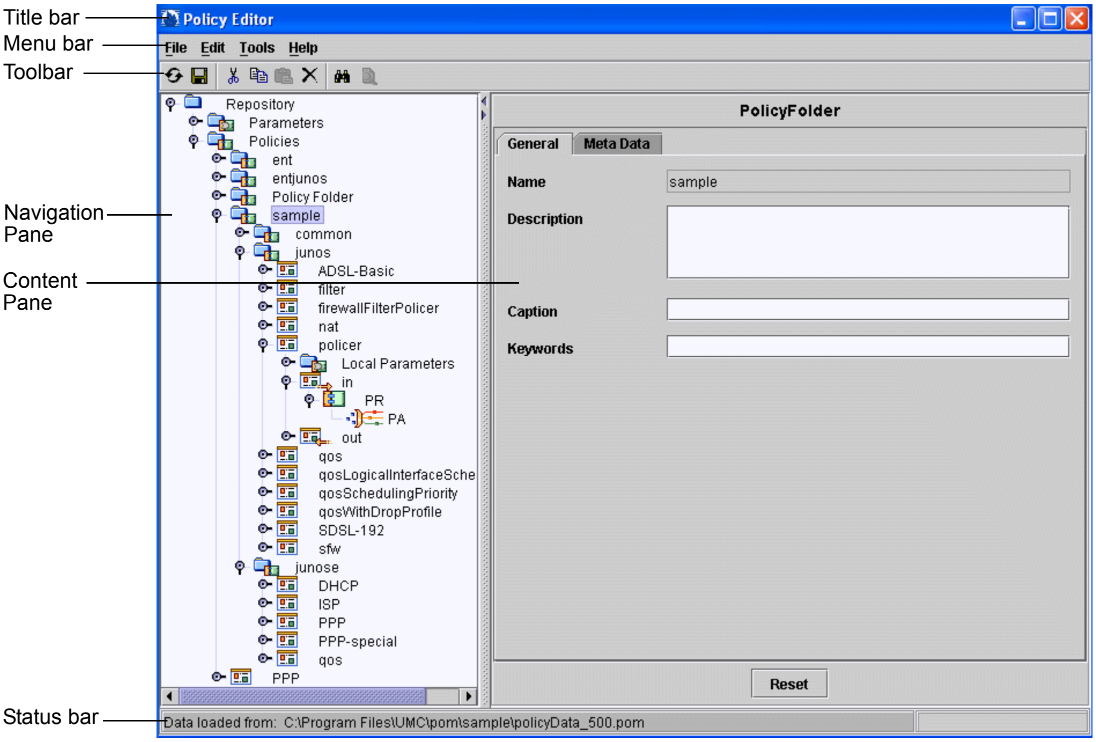
3. Click **OK**.

   The system connects you to the policy repository and displays the Policy Editor window.

Understanding the Policy Editor Layout

The Policy Editor window contains six main areas: title bar, menu bar, toolbar, navigation pane, content pane, and status bar (see Figure 19). The first five areas are the same ones described in *SRC-PE Getting Started Guide, Chapter 43, Using SDX Admin*.

Figure 19: Policy Editor Window Layout



Name: sample

Description:

Caption:

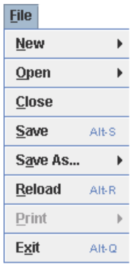
Keywords:

Data loaded from: C:\Program Files\UMC\pom\sample\policyData_500.pom

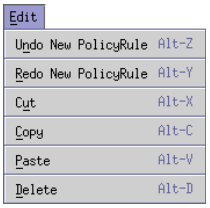
The status bar at the bottom of the Policy Editor window gives you additional information about Policy Editor. For example, the status bar can indicate the IP address of the policy repository to which you are connected.

Using the Menu Bar

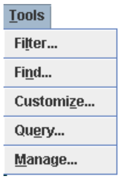
The menu bar allows you to execute commands related to each of the menus. See the following tables for information about each menu command.



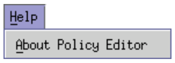
Menu	Command	Choose To
File	New	Open another Policy Editor main window or open a Policy Editor configuration file.
	Open	Open data source to get data into the GUI. You can open a connection to the directory host or open a file. For LDAP, the system prompts you to enter the requested directory connection parameters. Then the connection is made. For a file, you must choose a data file to be opened.
	Close	Close the connection to the directory or Close the current file. The policy objects in Policy Editor are cleared. **NOTE:** You receive a warning if you have not saved changes to the opened data source (directory or file).
	Save	Save changes to the directory or file. If the file did not previously exist, it is created. If the file exists, its contents are replaced with policies in Policy Editor.
	SaveAs	Save the policies in Policy Editor under a new file name. This in effect copies the policies to a new file. Be sure to give different versions of an object a unique name to avoid object name conflicts.
	Reload	Clear and reload policy objects in Policy Editor from the opened data source (directory or file). You receive a warning if there are pending changes that you have not saved to the opened data source (directory or file).
	Print	Print the file(s) you select.
	Exit	Exit the application. You receive a warning if there are pending changes that you have not saved to the opened data source (directory or file).



Menu	Command	Choose To
Edit	Undo	Cancel the most recent operation.
	Redo	Reinstate the operation that you cancelled with Undo.
	Cut	Cut the currently selected object.
	Copy	Copy the currently selected object.
	Paste	Paste the object copies from the cut or copy operation to the currently selected object.
	Delete	Delete the currently selected object.



Menu	Command	Choose To
Tools	Filter	Change the search filter used for the directory (see Table 21).
	Find	Find a specific instance of an object in Policy Editor.
	Customize	Customize the application based on your preferences.
	Query	Open the Router Query window. Use to find QoS profiles, policy groups, and routers.
	Manage	Allows you to access the CLIs of JUNOSe routers and JUNOS routing platforms.

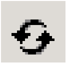
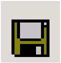
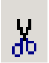
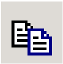
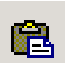
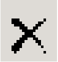
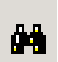
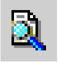


Menu	Command	Choose To
Help	About Policy Editor	View information about the Policy Editor software, including vendor name and software version.

Using the Toolbar

Table 17 shows the Policy Editor toolbar icons and the relationship between the icons and the menu commands. The toolbar icons exist in enabled and disabled modes. The mode depends on whether the operation is supported in the context of the selected object and on previous operations. For example, if no object was previously copied or cut, then Paste is disabled.

Table 17: Toolbar Functions

Icon	Corresponding Menu
	File > Reload
	File > Save
	Edit > Cut
	Edit > Copy
	Edit > Paste
	Edit > Delete
	Tools > Find
	Tools > Filter

Using the Navigation Pane

The navigation pane represents the policy management system in a tree format. The top-level folder is the root; the tree branches down to subfolders and then to individual objects.

The tree structure provides instance navigation and manipulation of the policy objects. However, not all objects are available in the navigation pane. For example, ProtocolCondition in the ClassifyTrafficCondition object is considered part of the ClassifyTrafficCondition object. Thus ProtocolCondition is not shown as a branch of ClassifyTrafficCondition in the navigation pane. ProtocolCondition is manipulated as part of the content pane for ClassifyTrafficCondition. For more information about navigating through ClassifyTrafficCondition objects, see *Using the Content Pane* on page 185.

Figure 20 shows the policy object hierarchy that is supported. All the connections between the objects are a parent-child relationship.

Figure 20: Policy Object Hierarchy in Navigation Pane

```
organizationFolder (logical root folder : Policy Repository)
   |-- organizationFolder (global folder : Parameters)
          |-- Parameter (global)
   |-- organizationFolder
          |-- policyGroup
          |-- organizationFolder
                 |-- policyGroup
                         |-- organizationFolder (logical folder: Local Parameters)
                                |-- Parameter (local)
                         |-- policyList
                                |-- policyRule
                                       |-- ClassifyTrafficCondition
                                       |-- FilterAction
                                       |-- ForwardAction
                                       |-- MarkAction
                                       |-- NextHopAction
                                       |-- NextInterfaceAction
                                       |-- RateLimitAction
                                       |-- TrafficClassAction
```

To manipulate objects in the navigation pane:

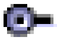
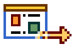
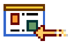
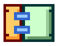
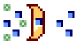
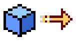
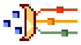
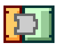
- Select an object—Click on the object. When you select an object, the object details appear in the content pane (see Table 19). From this pane you can add, change, or modify policy object parameters.
- Expand an object—Click on the expansion indicator icon to the left of the object (see Table 18). When the object is expanded, the icon points down. If there is no expansion indicator, then the object has been expanded to its lowest level.
- Collapse an object—Click on the expansion indicator icon to the left of the object. When the object is collapsed, the icon points toward the object.

After you have selected the object(s) you want, you can modify, add, cut, copy, paste, delete, or show messages for the object(s).

Navigation Pane Icons

Policy Editor uses different icons in the navigation pane to differentiate various object types under Policy Editor control. Table 18 shows and describes the icons used in the Policy Editor navigation pane.

Table 18: Policy Object Icons

Icon	Type	Description
	Expansion indicator	When indicator points down, folder is expanded
	Collapsed indicator	When indicator points toward the folder, folder is collapsed but can be expanded
	Repository folder	Repository folder is closed; you cannot see the contents
	Parameters folder	Folder containing local or global parameters
	Policies folder	Policy folder—Contains policy groups
	Valid policy group	Policy group—Contains valid policy lists
	Invalid policy group	Policy group—Contains invalid policy lists
	Ingress policy list	Ingress policy list—Contains ingress policy rules
	Egress policy list	Egress policy list—Contains egress policy rules
	Policy rule	Policy rule—Contains condition and action
	Classify	Condition object—Classify-traffic condition
	Forward	Action object—Forward action and NAT action
	Filter	Action object—Filter action and reject action
	Next hop	Action object—Next-hop action, routing instance action, and static routing instance action
	Next interface	Action object—Next-interface action
	Traffic mirror	Action object—Traffic mirror action
	Rate limit	Action object—Rate-limit action, policer action, traffic-shape action
	Mark	Action object—Mark action
	Traffic class	Action object—Traffic-class action, reference forwarding action, scheduler map action, loss priority action, FlowSpec action, DOCSIS action, and service class name action.
	Parameter	Parameter object with valid values
	Invalid parameter	Parameter object with invalid values

General Procedures for Using Policy Editor

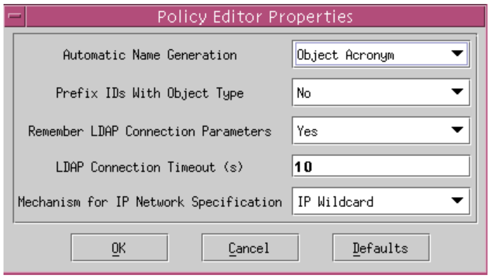
This section describes some general procedures that you can perform with Policy Editor.

Customizing Policy Editor Properties

The Customize command allows you to customize some of the properties in Policy Editor. To customize Policy Editor properties:

1. Click **Tools > Customize**.

   The Policy Editor Properties dialog box appears.



Policy Editor Properties

Automatic Name Generation	Object Acronym
Prefix IDs With Object Type	No
Remember LDAP Connection Parameters	Yes
LDAP Connection Timeout (s)	10
Mechanism for IP Network Specification	IP Wildcard

2. Select the parameters you want from the drop-down box in each field (see Table 19).

Table 19: Policy Editor Properties

Field	Options
Automatic Name Generation	Suggests the name base for the newly created object: ■ Object Type Based—Object type is used as prefix ■ Object Acronym—Object acronym is used as prefix ■ Anonymous—Word *anonymous* is used as prefix ■ None—No prefix is used
Prefix IDs With Object Type	Specifies whether or not object names in the navigation pane are prefixed with their object type. ■ Yes—Object names are prefixed with object type Example—PolicyGroup_fast ■ No—Object names are not prefixed with object type Example—Fast
LDAP Connection Timeout (s)	inactive
Remember LDAP Connection Parameters	■ Yes—System remembers the connection information ■ No—System does not remember the connection information
Mechanism for IP Network Specification	Choose between the following available IP address field labels: ■ IP Mask ■ IP Wildcard

Opening Multiple Policy Editor Windows

Clicking File > New Window allows you to open another main Policy Editor window. You can have multiple Policy Editor windows running at the same time. The status bar at the bottom of the window shows what the window is connected to.

You can copy and paste information from one window to another. You cannot cut and paste information from one window to another. If you use the drag-and-drop function (see *Using Drag and Drop to Cut and Paste Objects* on page 179), the operation will copy and paste information to the other window.

Printing Policy Objects

You can print the properties specified for a policy object or objects that you select either to a text file or to a printer. To print the information to a printer, you must have the printer configured to the system. (See your operating system or printer manual for configuration.)

The format of the print data is a structured representation of the object(s) you have selected.

You can print the properties from policy groups and policy folders; you cannot print the properties from a policy list, policy rule, policy condition, or policy action.

Undoing and Redoing Operations

Clicking Edit > Undo allows you to undo the last operation you performed on an object. Your most recent addition, modification, or deletion is canceled. Similarly, clicking Edit > Redo allows you to redo the last operation you performed on an object.

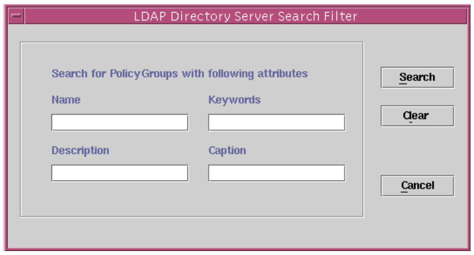
You can undo or redo up to 10 operations. However, you must undo or redo the most recent operation first. That is, if you want to undo a modification you made before you added an object, you must undo adding the object before you can undo your earlier modification.

When you click Undo or Redo, the name of the object type, not the name of the specific object, appears.

Filtering Searches

The Filter option allows you to limit the number of objects loaded in the navigation pane based on attributes of the policy group. Use the filter to group and identify objects that satisfy the search criteria. Filter is not active when you work with files.

Figure 21: Search Filter Window



LDAP Directory Server Search Filter

Search for PolicyGroups with following attributes

The filter is applied only to a search operation on the directory. It has no effect on newly created objects in the navigation pane or from the Save operation, because they do not result in an LDAP search operation.

Table 20 lists the supported attributes for the policy group search filter.

Table 20: LDAP Search Attributes

Filed Title	Attribute
Name	LDAP string for the policy group name
Keywords	LDAP string for keywords
Description	LDAP string for the description
Caption	LDAP string for the caption

Each entry creates a Boolean expression. For example, if you type **internet** in the Description field, the resulting Boolean expression is PolicyGroup.Description = *internet*. PolicyGroup.Description maps to the corresponding LDAP attribute name.

If you specify more than one field in the Filter window, the system combines the information in a logical AND operation.

Before changing the search filter, the system prompts you to save the changes made to the objects. You must save your changes because changing the filter results in cleaning the policy objects in the navigation pane and the Policy Editor directory cache.

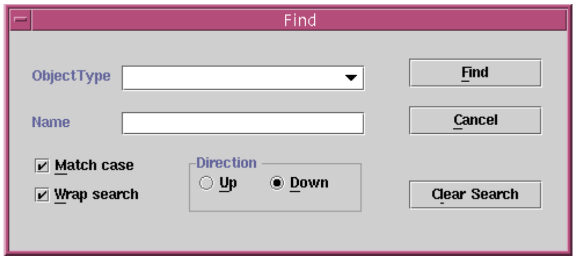
Finding Objects in the Navigation Pane

The Find command allows you to find objects in the navigation pane based on object type and name. The Find operation is started from the current selection in the navigation pane. If you have not selected an object in the navigation pane, Find starts from the root of the tree (that is, the Repository folder).

To find a policy object:

1. Click **Tools > Find**.

   The Find dialog box appears.



☑ Match case
☑ Wrap search
○ Up ● Down

2. Select an object type from the Object Type drop-down menu.

   Object Type is a mandatory field. If you do not select an object type and you start the Find operation, the system prompts you to enter an object type.

3. (Optional) Type the object name in the Name field.

4. Click **Find**.

You can refine your search as follows:

- Match case—Make the search case sensitive.
- Wrap search—Wrap back to the top of the search area when the end of the search area is reached.
- Direction—Find objects in the Up or Down directions with respect to the current selection in the navigation pane.

If the object cannot be found, the system displays a warning stating that the object matching the given criteria cannot be found.

If the object is found, select it. The system displays the information about the object in the content pane.

Running Queries for QoS Policy Information

See *SRC-PE Solutions Guide, Chapter 1, Managing Tiered and Premium Services with QoS on JUNOSe Routers with the SRC CLI*.

Accessing Router CLIs

You can use the Manage menu item to access the CLIs of JUNOSe routers and JUNOS routing platforms through a Telnet or SSH connection.

Modifying Policies

You can modify policies by using either the navigation or content pane.

To modify the information in the fields of a content pane, select the object (policy group, policy list, or policy rule), and then enter the data in the fields to be changed. The changes are saved after you click Save from the File menu.

Selecting Multiple Objects

You can perform operations (such as print, cut, or paste) on multiple objects that you select in the navigation pane. Highlight and select multiple objects by pressing Ctrl and clicking each object you want to select. The objects do not need to be in consecutive order. To select a group of objects in consecutive order, press Shift and click the consecutive objects you want to select.

To deselect an object, click on it again. It will no longer be highlighted.

Although you can select objects of different types, it is not useful to do so. Some, but not all, operations can be performed concurrently on multiple objects that are of different types.

Using Drag and Drop to Cut and Paste Objects

You can cut and paste selected objects by holding down the left mouse key and dragging the object or objects to a new place in the navigation pane. If an object you have re-placed has the same name as an object already in the folder, you are prompted to rename it.

Cutting Objects

Policy Editor places information that you cut into a clipboard for future use. To cut a policy object:

1. Highlight the policy group, policy list, or policy rule, and right-click.

2. Click **Cut**.

If you cut a policy group, policy list, or policy rule, the system deletes any child objects and references that exist.

Copying Objects

When you copy an object, the system leaves the information in the original location and copies it to the clipboard for future use. To copy a policy object:

1. Highlight the policy group, policy list, or policy rule you want to copy, and right-click.
2. Click **Copy**.

Be sure to change the policy group, policy list, or policy rule child objects and references.

Pasting Objects

Pasting a policy group, policy list, or policy rule object means copying the object from the clipboard to a new folder.

To paste a policy object:

1. Highlight the object that you want to paste the information into, and right-click.
2. Click **Paste**.

Be sure to give the pasted policy group, policy list, or policy rule a unique name. If the object itself has child objects, the child object names are present. Policy Editor assists you with the name for the pasted object; the system either preserves the name of the original object or suggests a different one to avoid name conflict.

Deleting Policy Objects

To delete a policy object:

1. Highlight the policy object, and right-click.
2. Click **Delete**. The Deleting dialog box appears.
3. Click **OK** to delete the policy object. Click **Cancel** to close the dialog box without deleting the policy object.

NOTE: If you change or delete an item, make sure that you change or delete all of its dependencies.

Reloading a Policy Object

Reload gets the last saved copy of the information. All the information that you entered since the last save and before you execute the reload command is lost.

If you delete a policy object and then decide that you want to keep it:

1. Select **File**, and click **Reload**. The Reloading from Data Source dialog box appears.

2. Click **No**, and the element or object is not deleted.

NOTE: If you save the policy, the deleted information is lost.

Using Pop-Up Menus

If you select a folder or individual object in the navigation pane and then right-click, a pop-up menu appears. Available commands relative to the selected object appear. If the command appears dimmed, it is not available.

Figure 22 is an example of an object pop-up menu.

Figure 22: Sample Object Pop-Up Menu

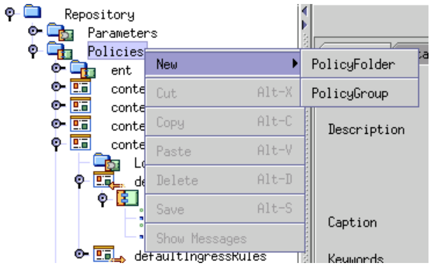


Table 21 lists the menu selections that are available from the various pop-up menus.

Table 21: Policy Editor Pop-Up Menus

Object Selected	Menu Item	Description
Parameters folder	New	New policy group.
		New organizational folder.
		New parameter. Available for organizational folder for global and local parameters folder.
	Cut	Cuts selected object. Cut is not allowed on the top folder, which is the base DN.
	Copy	Copies selected object.
	Paste	Pastes previously copied or cut object in this container, when available.
	Delete	Deletes selected object. Delete is not allowed on the top folder, which is the base DN.
	Save	
	Show Messages	Displays transient messages that are related to Policy Editor knowledge of the selected object. The messages can include information such as validation error messages or the modification status of the object in Policy Editor. (For example, does it require a Save, or is it up to date?) If there are no messages, the menu item is not available.
Parameter	New	
	Cut	Cuts selected object.
	Copy	Copies selected object.
	Paste	Pastes previously copied or cut object in this container. If there are no previously copied or cut objects, the menu item is not available.
	Delete	Deletes selected object.
Policy Folder	New	New policy folder or policy group.
	Cut	Cuts selected object. Cut is not allowed on the top folder, which is the base DN.
	Copy	Copies selected object.
	Paste	Pastes previously copied or cut object in this container. If there are no previously copied or cut objects, the menu item is not available.
	Delete	Deletes selected object. Delete is not allowed on the top folder, which is the base DN.
	Save	
	Show Messages	Displays transient messages that are related to Policy Editor knowledge of the selected object. The messages can include information such as validation error messages or the modification status of the object in Policy Editor. (For example, does it require a Save, or is it up to date?) If there are no messages, the menu item is not available.

Table 21: Policy Editor Pop-Up Menus (continued)

Object Selected	Menu Item	Description
Policy Group	New	New JUNOS or JUNOSe policy list.
	Cut	Cut selected object.
	Copy	Copies selected object.
	Paste	Pastes previously copied or cut object in this container. If there are no previously copied or cut objects, the menu item is not available.
	Delete	Deletes selected object.
	Save	
	Show Messages	Displays transient messages that are related to Policy Editor knowledge of the selected object. The messages can include information such as validation error messages or the modification status of the object in Policy Editor. (For example, does it require a Save, or is it up to date?) If there are no messages, the menu item is not available.
Policy List	New	New policy rules are displayed relative to the selected object.
	Cut	Cuts selected object.
	Copy	Copies selected object.
	Paste	Pastes previously copied or cut object in this container. If there are no previously copied or cut objects, the menu item is not available.
	Delete	Deletes selected object.
Policy Rule	New	New conditions and actions are displayed relative to the selected object.
	Cut	Cuts selected object.
	Copy	Copies selected object.
	Paste	Pastes previously copied or cut object in this container. If there are no previously copied or cut objects, the menu item is not available.
	Delete	Deletes selected object.
Any condition		
Any action	New	Selection is not available.
	Delete	Deletes selected object.
	Cut	Cuts selected object.
	Copy	Copies selected object.
	Paste	Selection is not available.

Using the Content Pane

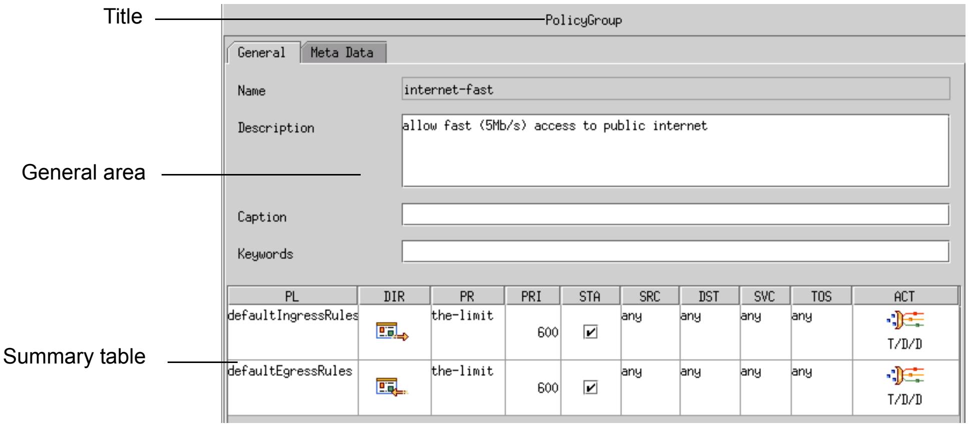
The content pane (see Figure 19) lists the details of a policy object. *Chapter 6, Policy Management Overview* contains samples of Policy Editor content panes for the various objects.

The content pane is used for access to object details and the modification of policy objects. The information in the Meta Data tab represents creation and modification timestamps for the objects. For policy group, policy list, and policy rule objects, the General tab in the content pane consists of two subsections: a general area and a summary table:

- General area—Shows general information about the object. You are provided with various mechanisms to update data, such as captions and descriptions.
- Summary table—Shows information related to the object in a summary form.

Figure 23 shows a PolicyGroup pane with its general area and summary table.

Figure 23: Content Pane Sections—General Area and Summary Table



Name	internet-fast
Description	allow fast (5Mb/s) access to public internet
Caption	
Keywords	

PL	DIR	PR	PRI	STA	SRC	DST	SVC	TOS	ACT
defaultIngressRules		the-limit	600	☑	any	any	any	any	T/D/D
defaultEgressRules		the-limit	600	☑	any	any	any	any	T/D/D

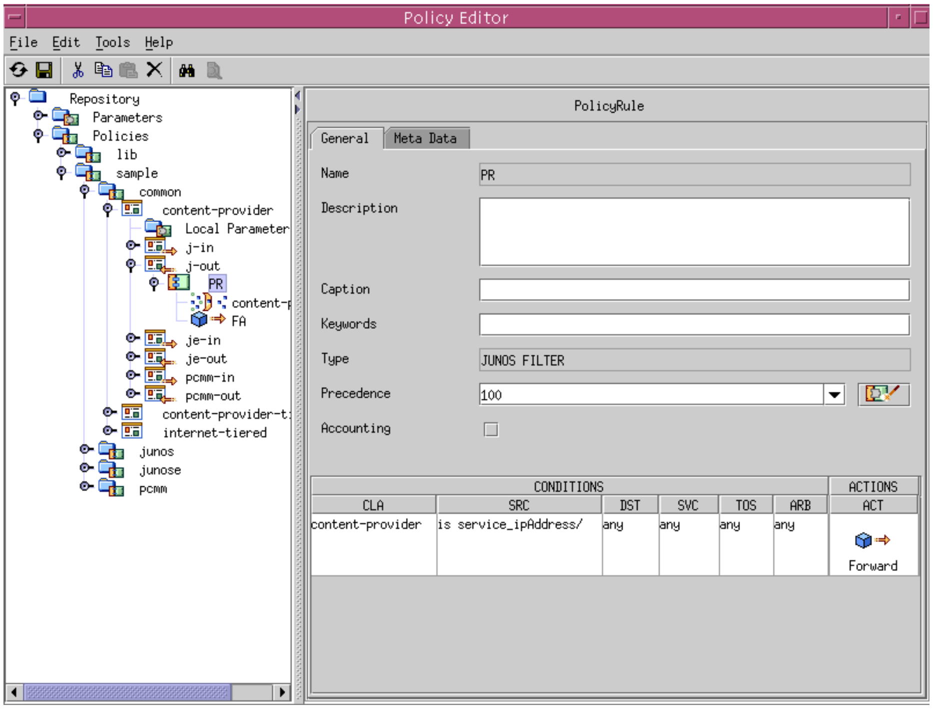
You can update the data in the summary table itself or from a dialog box that appears when you double-click on an object in the summary table. If you right-click within a cell, a pop-up menu appears from which you can perform additional actions.

You can also double-click on a policy list or policy rule object in the summary table to select an object in the navigation pane.

The content pane changes based on your interaction to show only relevant information.

You can also modify values for conditions and actions by selecting a policy rule object in the navigation pane. The summary table in the General tab of the PolicyRule pane contains five columns, as shown in Figure 24.

Figure 24: Condition and Action Columns in PolicyRule Pane



Name: PR

Description:

Caption:

Keywords:

Type: JUNOS FILTER

Precedence: 100

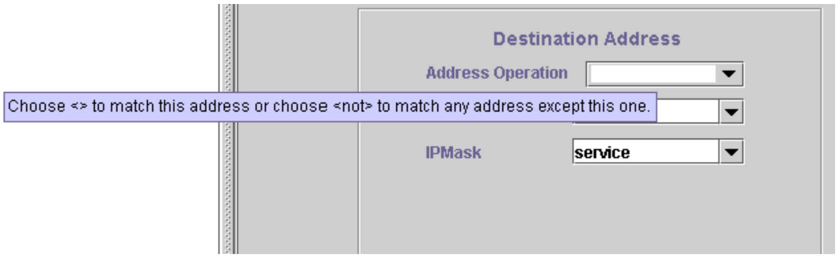
Accounting ☐

CONDITIONS						ACTIONS
CLA	SRC	DST	SVC	TOS	ARB	ACT
content-provider	is service_ipAddress/	any	any	any	any	Forward

Using Tool Tips

Tool tips provide information about toolbar buttons, the fields in a pane, and objects in the navigation pane. Tool tips are provided only for buttons, fields, or objects that can benefit from the additional information. To view tool tips, place the cursor over a button, field, or object. The tool tip appears (for example, see Figure 25).

Figure 25: Tool Tip for Address Operation



Choose <> to match this address or choose <not> to match any address except this one.

IPMask: service

Internationalization

Policy Editor supports translating messages (strings), such as label texts, error messages, menu items, and dialogs into local languages.

External property files are provided so that a knowledgeable operator can customize these user-visible strings into local languages without altering the software code. In a property file, each externalized string variable is stored as an entry that is part of a key-value pair.

The key is a meaningful representation of a specific string variable; for example, RateLimitActionPanel.PeakRate.label. The value is a customized part of the string content; for example, Peak Rate (bps).

The following shows several examples of externalized strings with customized parts of the string content:

```
. . .
RateLimitActionPanel.PeakRate.label=PeakRate(bps)
RateLimitActionPanel.PeakBurst.label=PeakBurst(bytes)
RateLimitActionPanel.CommittedAction.label=CommittedAction
. . .
```

This feature provides different user-visible contents in Policy Editor based on the customer's location. With the same code, but with different external property files, various users can see different visual representations of the same string.

Some constraints are applied to the external properties to ensure proper layout in Policy Editor panes. In the preceding example, a constraint is placed on the maximum length of the label.

For example, if the maximum length of the label is 39 characters but the actual label length is 200 characters, Policy Editor uses the first 39 characters of the label and logs an error in the log file.

Table 22 lists files that contain externalized strings. They are located in the directory *<pom-install-dir>/etc*.

Table 22: Files with Externalized Strings

File Name	Purpose
HelpDialog.properties	Help
MenuAttributes.properties	Menus
PanelAttributes.properties	Panels
FindDialog.properties	Find dialog
CustomizationAttributes.properties	Customize dialog
LDAPConnectionDialog.properties	LDAP connection dialog
FilterDialog.properties	Filter dialog

Storing and Retrieving Policies

In addition to the directory, Policy Editor lets you use store policies in and retrieve policies from the file system where Policy Editor is running. The policy engine uses only the policies in the directory. The policies in the files are for your use locally and are not visible to the policy engine.

A connection to the directory is required for file operations. The file format supported is internal to Policy Editor. It is not a public, open file format.

NOTE: You must not manually edit the file by using another editor. This operation can cause corrupted files or version problems in the files.

Sorting Objects

The objects in object collection views are sorted in case-insensitive nonlocale lexicographical order using Unicode with the object name/id.

The following logic is used for comparing two strings. It is specified by String.compareToIgnoreCase in Java:

1. Two strings are compared lexicographically, ignoring case and locale. The system achieves this operation by converting each string to uppercase and then lowercase before performing the lexicographical string comparison.

2. This method returns an integer whose sign is that of the following:

   this.toUpperCase().toLowerCase().compareTo(str.toUpperCase().
   toLowerCase())

NOTE: This sorting logic does not take locale into account and results in an unsatisfactory ordering for certain locales (for example, Turkey).

3. The sort is applied to every level in a hierarchy for objects in that level when object collection is part of the hierarchy. For example, the policy group folder can contain several policy list objects. The objects are sorted alphabetically using the policy list name for comparison.

4. The object type or other attributes can take higher precedence over the name sort. For example, the precedence can be (in order of highest to lowest) direction, condition, actions, sorted name. This order sorts the group by functionality. It is used in the multicolumn sort operations. Within each group the objects are sorted. In addition, logical ordering can exist, which takes precedence over alphabetic ordering.

Chapter 8

Overview of Using Local and Global Parameters

This chapter provides an overview of using local and global parameters in policies. Topics include:

- Overview of Global and Local Parameters on page 189
- Parameter Types on page 190

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. For information about parameter value acquisition, see *Generating Policies by Specifying Parameters on page 395*.

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.

Parameter Types

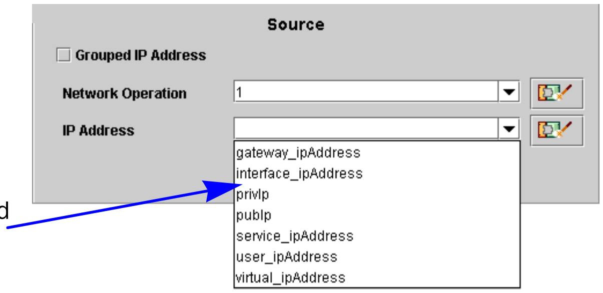
Global and local parameters are assigned a type. (Note that the term *type* is used in the SRC CLI and Policy Editor, and the term *role* is used in SDX Admin. Both terms have the same meaning.) The type indicates in which options in the SRC CLI or which Policy Editor fields you can use the parameter.

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
```

In Policy Editor, wherever there is a field for which you can specify an address, a drop-down list displays all the global parameters of type address as well as local parameters of type address that are defined in the policy group in which you are working. Figure 26 shows a drop-down list of global and local parameters for parameter type address.

Figure 26: Drop-Down List of Local and Global Address Parameters in Policy Editor



Source

□ Grouped IP Address

Network Operation: 1

IP Address

Global and local parameters defined for parameter type address

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 23 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 23: Parameter Types (or Roles)

Type	Predefined Parameters	Used In	Used to Specify
address	gateway_ipAddress interface_ipAddress service_ipAddress user_ipAddress virtual_ipAddress	Classify-traffic condition Next-interface action Next-hop action	IP addresses in dotted decimal notation.
addressMask	interface_ipMask service_ipMask user_ipMask	Classify-traffic condition	IP masks in dotted decimal notation. For JUNOSe 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. Policy Editor 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 percent	Policer action	
boolean	false true		
burst		Rate-limit action Policer action DOCSIS action	Burst sizes. The range is 2^{14}—2^{32}–1.
color	green red yellow	Classify-traffic condition Color action	Color of action or classifier. The policy engine and Policy Editor validate these values; the substitution engine does not.
dceRpcUuid		Classify-traffic condition	
dropProfileProtocol	any_protocol non_tcp tcp_only	Scheduler action	
dropProfileType	interpolated segmented	Scheduler action	
forwardingClass		Classify-traffic condition QoS condition	

Table 23: Parameter Types (or Roles) (continued)

Type	Predefined Parameters	Used In	Used to Specify
fragOffset		Classify-traffic condition	The value of the fragment offset field of IP packets. For JUNOSe routers: ■ eq 0—Equal to 0 ■ eq 1—Equal to 1 ■ gt 1—Greater than 1 ■ any—Any value For JUNOS routing platforms and PCMM policies, integer in the range 0–8191. The policy engine and Policy Editor validate these values; the substitution engine does not.
grantSize		DOCSIS action	
icmpCode icmpType		Classify-traffic condition	8-bit values that represent patterns in the ICMP code and ICMP type fields in IP packets. The policy engine and Policy Editor validate these values; the substitution engine does not.
igmpType		Classify-traffic condition	8-bit values that represent patterns in the IGMP type field in IP packets. The policy engine and Policy Editor validate these values; the substitution engine does not.
interfaceGroup		Classify-traffic condition	
InterfaceSpec	bfwIf gfwIf	Next-interface action	The router interface. For JUNOSe 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 ipFlagsMask		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.
ipSecSpi		Classify-traffic condition	
IPv4range			
jitter		DOCSIS action	
matchDirection	both input output	Classify-traffic condition	
maxLatency		DOCSIS action	
messageType		Reject action	

Table 23: Parameter Types (or Roles) (continued)

Type	Predefined Parameters	Used In	Used to Specify
microSecond			
natTranslationType		NAT action	
network	any	Classify-traffic condition NAT action	IP subnets using two forms: <address>/<mask> <address>/<prefixLength> where <address> and <mask> are in the traditional dotted decimal notation. <prefixLength> is a number in the range 0–32, which specifies how many of the first bits in the address specify the network. 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.
networkOperation		Classify-traffic condition	Whether a network field of a packet should match or not match the value specified in a policy condition. ■ 0—Does not match ■ 1—Matches
packetLength		Classify-traffic condition DOCSIS action FlowSpec action	
packetLossPriority	any_priority high_priority low_priority	Loss priority action	
packetOperation		Rate-limit action Policer action Stateful firewall	Actions taken on packets. For rate-limit actions, valid values are: $'forward', $'filter', and $'mark <tosByte> <tosMask>'. For policer actions, valid values are: filter, forwardingClass, lossPriority. For stateful firewalls, valid values are: filter, forward, reject. The policy engine and Policy Editor validate these values; the substitution engine does not.
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.

Table 23: Parameter Types (or Roles) (continued)

Type	Predefined Parameters	Used In	Used to Specify
portOperation	eq neq	Classify-traffic condition	Whether a port field should match or not match the value(s) specified in a condition. For JUNOSe policies valid values are: $'eq', $'lt', $'gt', $'neq' and $'range'. For JUNOS the allowed values are: ■ 0—Does not match ■ 1—Matches The policy engine and Policy Editor validate these values; the substitution engine does not.
prPrecedence		Policy rule	
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 and Policy Editor validate 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. ■ 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 JUNOSe 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 and Policy Editor validate 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 23: Parameter Types (or Roles) (continued)

Type	Predefined Parameters	Used In	Used to Specify
schedulerPriority	high low medium_high medium_low strict_high	Scheduler action	
schedulerTransmitRate		Scheduler action	
schedulerTransmitRateUnit	rate_in_bps rate_in_percentage rate_in_remainder	Scheduler action	
serviceClassName		Service class name action	
serviceNumber	controlled_load_service guaranteed_service	FlowSpec action	
sessionClassIdPriority		GateSpec action	
slackTerm		FlowSpec action	
snmpCommand	get get_next set trap	Classify-traffic condition	
tcpFlags tcpFlagsMask		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.
timeout		Classify-traffic condition	
tokenBucketSize		FlowSpec action	
tosByte tosByteMask		Classify-traffic condition Rate-limit action Mark action	8-bit values that represent patterns in the ToS byte field in IP packets. When tosByteMask is used in ToS conditions, the allowed values are 0, 224, 252, and 255. The policy engine and Policy Editor validate 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 JUNOSe router.
trafficPriority		DOCSIS action	

Table 23: Parameter Types (or Roles) (continued)

Type	Predefined Parameters	Used In	Used to Specify
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 JUNOSe IPv6 policies, valid values are in the range 0–15. The policy engine and Policy Editor validate these values; the substitution engine does not.

Predefined Global Parameters

Table 24 describes the predefined built-in and runtime global parameters that the SRC software provides. Only three of the predefined parameters can be modified: any, bfwlf, and gfwlf.

Table 24: 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	
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	

Table 24: Predefined Global Parameters (continued)

Predefined Parameter	Description	Type	Runtime
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
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
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	
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	
netbios	Specifies the NetBIOS protocol	applicationProtocol	
netshow	Specifies the NetShow protocol	applicationProtocol	

Table 24: Predefined Global Parameters (continued)

Predefined Parameter	Description	Type	Runtime
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	
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
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	

Table 24: Predefined Global Parameters (continued)

Predefined Parameter	Description	Type	Runtime
two_rate	Sets the rate-limit type to two rate	rateLimitType	
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.

When you perform a SaveAs operation to a directory with Policy Editor, the SRC software does not verify the names of local parameters in the policy group with the names of existing global parameters in the directory. After the SaveAs operation is complete, the directory may contain global parameters and local parameters with the same names. You will not receive any messages about duplicate names. If local and global parameters have duplicate names, the policy engine uses the local parameter definitions.

Chapter 9

Configuring Local and Global Parameters with Policy Editor

This chapter describes how to configure global and local parameters with Policy Editor. You can also use the following to configure parameters:

- To use the SRC CLI, see *Chapter 10, Configuring and Managing Policies with the SRC CLI*.
- To use the C-Web interface, see *SRC-PE C-Web Interface Configuration Guide, Chapter 22, Configuring Local and Global Parameters with the C-Web Interface*.

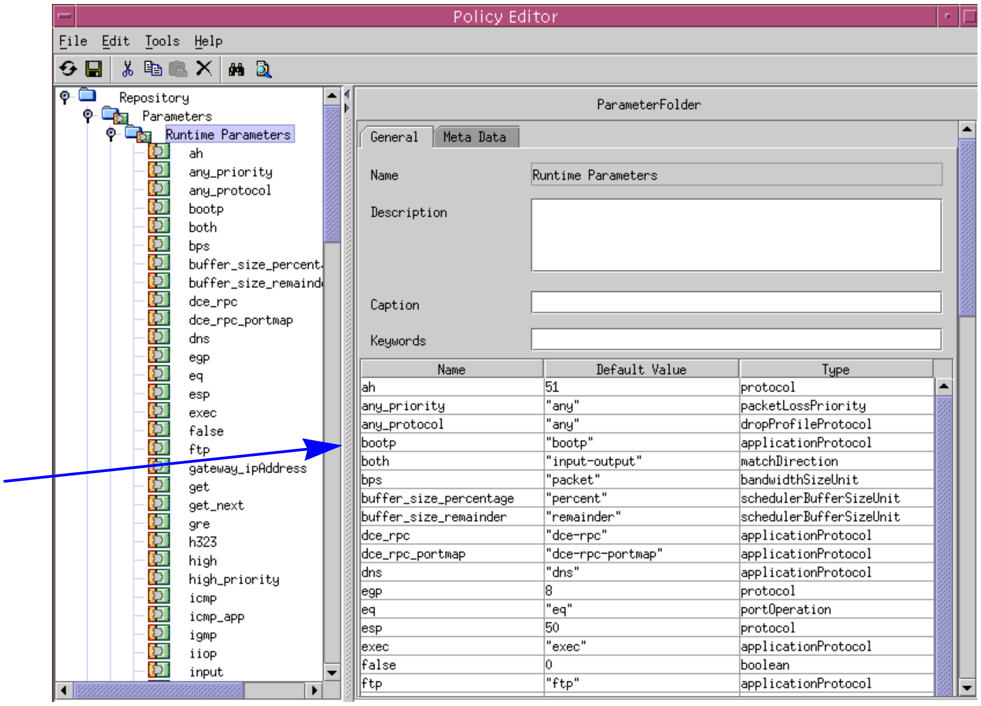
Topics in this chapter include:

- Viewing Global Parameters in Policy Editor on page 201
- Viewing Local Parameters in Policy Editor on page 202
- Creating and Modifying Global Parameters in Policy Editor on page 203
- Creating and Modifying Local Parameters in Policy Editor on page 206

Viewing Global Parameters in Policy Editor

In Policy Editor, global parameters are held in the Parameters repository. To view global parameters, expand and select the Parameters folder in the Policy Editor navigation pane. In Figure 27, the Runtime Parameters folder is selected, which causes all parameters in that folder to appear in the parameter summary table. To view a specific parameter, select the parameter in the navigation pane.

Figure 27: Parameter Folder with Global Parameters



Name: Runtime Parameters

Description:

Caption:

Keywords:

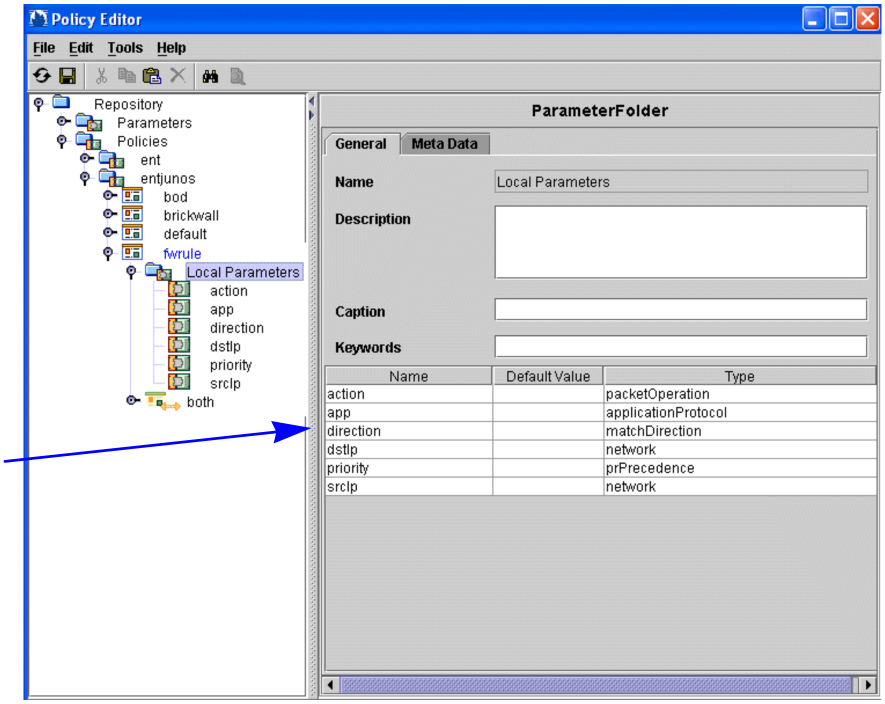
Name	Default Value	Type
ah	51	protocol
any_priority	"any"	packetLossPriority
any_protocol	"any"	dropProfileProtocol
bootp	"bootp"	applicationProtocol
both	"input-output"	matchDirection
bps	"packet"	bandwidthSizeUnit
buffer_size_percentage	"percent"	schedulerBufferSizeUnit
buffer_size_remainder	"remainder"	schedulerBufferSizeUnit
dce_rpc	"dce-rpc"	applicationProtocol
dce_rpc_portmap	"dce-rpc-portmap"	applicationProtocol
dns	"dns"	applicationProtocol
egp	8	protocol
eq	"eq"	portOperation
esp	50	protocol
exec	"exec"	applicationProtocol
false	0	boolean
ftp	"ftp"	applicationProtocol

Summary table of all global parameters in Runtime Parameters folder

Viewing Local Parameters in Policy Editor

When you create a policy group, Policy Editor automatically creates a Local Parameters folder inside the policy group. To view all local parameters for a policy group, select the Local Parameters folder in the navigation pane. (See Figure 28.) All local parameters that you create within the policy group are displayed in a summary table in the ParameterFolder pane. To view a specific parameter, select the parameter in the navigation pane.

Figure 28: Local Parameter Folder



Name: Local Parameters

Description:

Caption:

Keywords:

Name	Default Value	Type
action		packetOperation
app		applicationProtocol
direction		matchDirection
dstIp		network
priority		prPrecedence
srcIp		network

Summary table of all local parameters in the Local Parameters folder

(Note that the term *type* is used in Policy Editor, and the term *role* is used in SDX Admin. Both terms have the same meaning.)

Creating and Modifying Global Parameters in Policy Editor

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.

You can create a global parameter from the Parameters folder and from within a policy.

Creating Global Parameters from the Parameters Folder

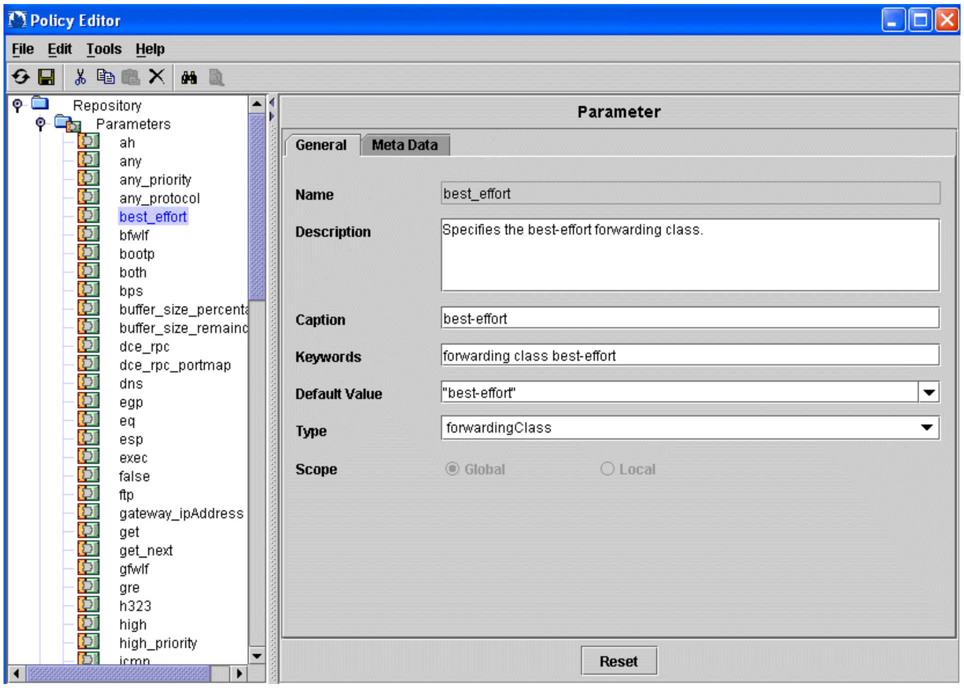
To create a global parameter from the Parameters folder:

1. In the Policy Editor navigation pane, right-click the **Parameters** folder, and select **New > Parameter**.

2. In the Parameter Name dialog box, assign a name to the parameter. This is the name that appears in Policy Editor drop-down menus.

3. Select the new parameter in the navigation pane.

   The Parameter pane appears.



Name: best_effort

Description: Specifies the best-effort forwarding class.

Caption: best-effort

Keywords: forwarding class best-effort

Default Value: "best-effort"

Type: forwardingClass

Scope: ● Global ○ Local

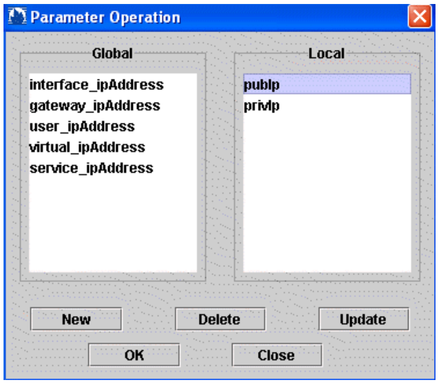
4. Fill in the fields as described in *Parameter Definition Fields* on page 205.

Creating Global Parameters Within a Policy

Policy fields for which you can use parameters have a ✎ icon next to them. To create a global parameter from within a policy:

1. Click the ✎ icon next to the field for which you want to create a global parameter.

   The Parameter Operation dialog box appears. Existing local and global parameters are listed. You can add, delete, or modify global parameters from this dialog box. You can also add local parameters.

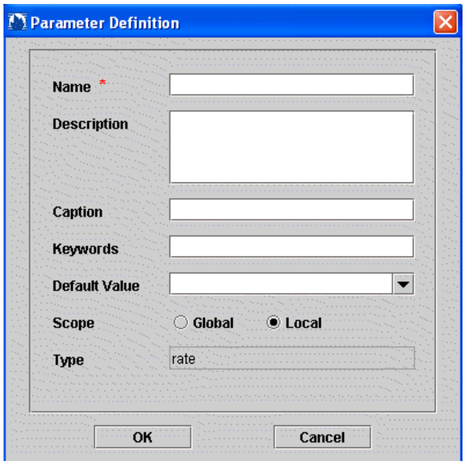


Global: interface_ipAddress, gateway_ipAddress, user_ipAddress, virtual_ipAddress, service_ipAddress

Local: publp, privlp

2. Click **New**.

The Parameter Definition dialog box appears.



Parameter Definition

Name *

Description

Caption

Keywords

Default Value

Scope ○ Global ● Local

Type rate

3. Fill in the fields as described in *Parameter Definition Fields* on page 205.

Parameter Definition Fields

Use the fields in this section to configure parameters.

Description

- Description of the parameter. You can provide extra information and examples of how the parameter is used.
- Value—Text
- Default—No value

Caption

- Short description of the parameter.
- Value—Text
- Default—No value

Keywords

- Keywords that you can use to search for a parameter.
- Value—Text
- Default—No value

Default Value

- An optional value that the policy engine uses if no other values are provided during the parameter value acquisition process. If other values are provided to the policy engine but problems are encountered, the default value for the parameter is not used. The policy engine generates an error.
- Value—Valid value for the parameter type; see Table 23 on page 191 for valid values of each parameter type
- Default—No value

Scope

- Specifies whether the parameter is a global parameter or a local parameter.
- Value
  - global—Select to create a global parameter.
  - local—Select to create a local parameter.
- Default—Local

Type

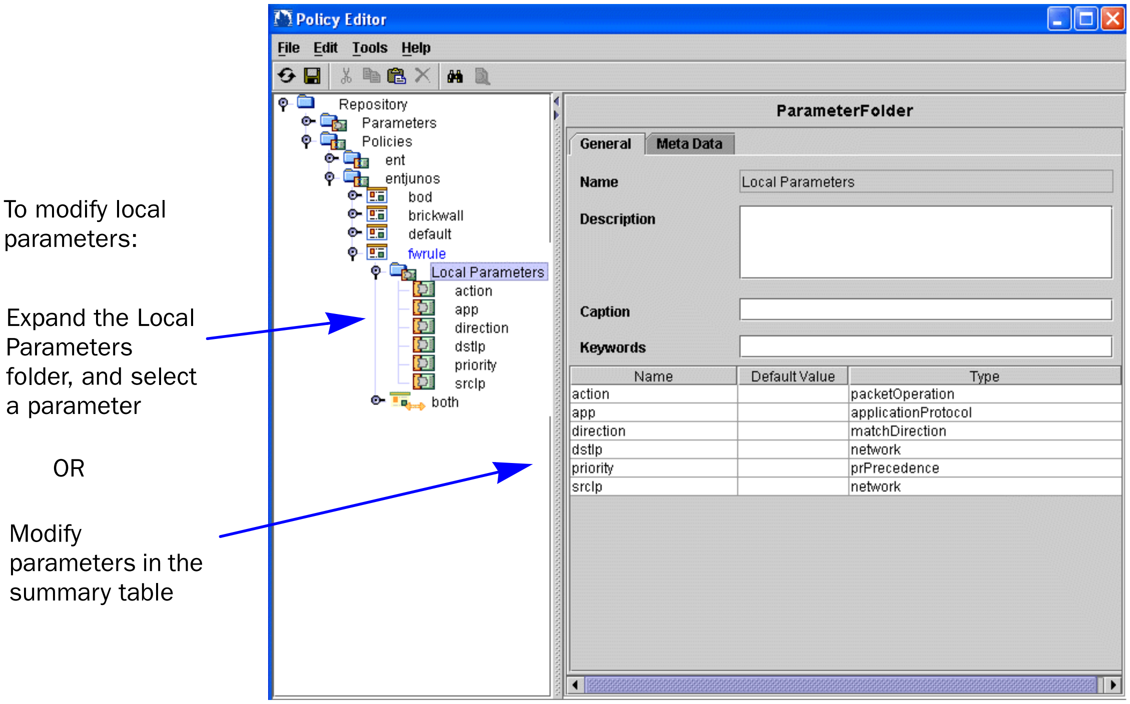
- Type of attribute for which you can use the parameter. The parameter type determines where you can use the parameter.
- Value—Select from drop-down list. See Table 23 on page 191 for a description of parameter types.
- Default—No value

Creating and Modifying Local Parameters in Policy Editor

When you create a policy group, Policy Editor automatically creates a Local Parameters folder inside the policy group. (See Figure 29.) All local parameters that you create within the policy group are added as objects in the Local Parameters folder. The ParameterFolder pane also displays a summary table of all local parameters in the policy group.

When you perform a SaveAs operation to a directory server, the SRC software does not check the names of local parameters in the policy group with names of existing global parameters in the directory server. After the SaveAs operation is complete, the directory server may contain global parameters and local parameters with the same names. You will not receive any messages about duplicate names. If local and global parameters have duplicate names, the policy engine uses the local parameter definitions.

Figure 29: Modifying Local Parameters



Name: Local Parameters

Description:

Caption:

Keywords:

Name	Default Value	Type
action		packetOperation
app		applicationProtocol
direction		matchDirection
dstIp		network
priority		prPrecedence
srcIp		network

To modify local parameters:

Expand the Local Parameters folder, and select a parameter

OR

Modify parameters in the summary table

To modify parameters from the summary table:

- Double-click on a parameter name entry to display the Parameter content pane. Fill in the fields as described in *Parameter Definition Fields* on page 205.
- Type a new value in a Default Value field.
- Click on an entry in the Type column to display a drop-down list of types.

Creating a Local Parameter

There are two ways to create a local parameter.

From the Local Parameters folder:

1. Right-click the **Local Parameters** folder, and select **New > Parameter**.
2. In the Parameter Name Dialog box, assign a name to the parameter. This is the name that appears in Policy Editor drop-down menus.

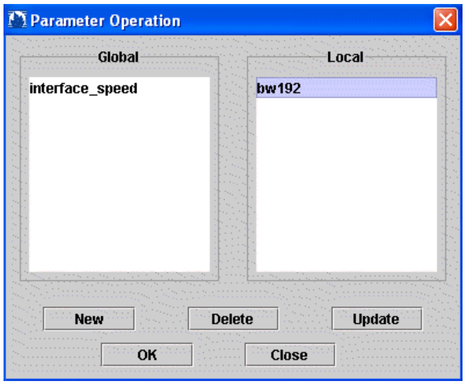
   The Parameter pane appears.
3. Fill in the fields as described in *Parameter Definition Fields* on page 205.

Policy fields for which you can use parameters have a ✎ icon next to them. To create a local parameter from within a policy:

1. Click the ✎ icon next to the field for which you want to create a local parameter.

   The Parameter Operation dialog box appears.

   Existing local and global parameters are listed. You can add, delete, or modify local parameters from this dialog box. You can also add global parameters.

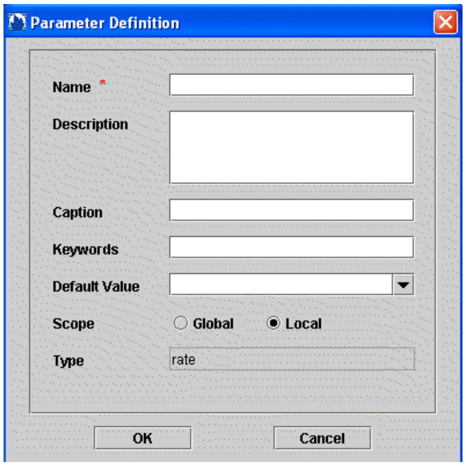


Global: interface_speed

Local: bw192

2. Click **New**.

   The Parameter Definition dialog box appears.



Scope ○ Global ● Local

Type rate

3. Fill in the fields as described in *Parameter Definition Fields* on page 205.

Chapter 10

Configuring and Managing Policies with the SRC CLI

This chapter describes how to use the SRC CLI to configure and manage policies. You can also use the following to configure and manage policies:

- To use the C-Web interface, see *SRC-PE C-Web Interface Configuration Guide, Chapter 21, Configuring and Managing Policies with the C-Web Interface***.**

- To use Policy Editor, see *Chapter 11, Configuring and Managing Policies with Policy Editor*.

Topics in this chapter include:

Before You Configure 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
3. 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* on page 404.

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:

```
...
JunoseIpPolicyClaclRuleEntry ::= SEQUENCE {
...
junoseIpPolicyClaclRuleTosByte Integer32,
junoseIpPolicyClaclRuleTosMask 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.

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
```

Configuring Policy Folders

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
  ```

Configuring Policy Groups

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";
   ```

Configuring Policy Lists

When you add a policy list, you specify whether the policy list is for JUNOS routing platforms (junos), JUNOSe routers (junose-ipv4, junose-ipv6), or a CMTS device (pcmm). 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);
    applicability (input | output | both);
    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.

   ```
   [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 JUNOSe DHCP";
   ```

Configuring 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 JUNOSe policy lists and PCMM policy lists. 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.

Adding a Policy Rule

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. (0ptional) 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";
   ```

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. See:
   - Creating a Classify-Traffic Condition on page 220
2. Configure source networks. You can configure source networks in one of two formats. See:
   - Configuring Source Networks on page 221
   - Configuring Source Grouped Networks on page 222
3. Configure destination networks. You can configure destination networks in one of two formats. See:
   - Configuring Destination Networks on page 223
   - Configuring Destination Grouped Networks on page 224
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 on page 225
   - To configure protocol conditions that include ports, see:
     - Configuring Protocol Conditions with Ports on page 226
   - To configure protocol conditions in which the protocol that you specify is a parameter, see:
     - Configuring Protocol Conditions with Parameters on page 229
   - To configure protocol conditions in which the protocol is TCP, see:
     - Configuring TCP Conditions on page 233
   - To configure protocol conditions in which the protocol is ICMP, see:
     - Configuring ICMP Conditions on page 235
   - To configure protocol conditions in which the protocol is IGMP, see:
     - Configuring IGMP Conditions on page 237

- To configure protocol conditions in which the protocol is IPSec, see:
  - Configuring IPSec Conditions on page 238
- To configure a ToS byte condition, see:
  - Configuring ToS Byte Conditions on page 239

5. For JUNOS filter policies, configure a JUNOS filter condition. See:
   - Configuring JUNOS Filter Conditions on page 240
6. For the stateful firewall and NAT policies, configure an application protocol condition. See:
   - Configuring Application Protocol Conditions on page 241

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.

Before You Configure Classify-Traffic Conditions

If you are configuring classifiers for PCMM policies, you can specify whether the classifier will be used in a PCMM IO2 or IO3 network. By default, the software translates classify-traffic conditions into PCMM IO2 classifiers.

- See *Specifying the PCMM Classifier Type* on page 218.

For JUNOSe policies, you can specify that the SAE expand the classifier into multiple classifiers before it installs the policy on the router.

- See *Enabling Expansion of JUNOSe Classify-Traffic Conditions* on page 218.

Enabling Expansion of JUNOSe Classify-Traffic Conditions

For information about expanded classifiers, see *Expanded Classifiers* on page 153.

Use the following configuration statement to enable or disable the expansion of JUNOSe classifiers.

```
shared sae configuration policy-management-configuration {
    enable-junose-classifier-expansion;
}
```

To enable or disable the expansion of JUNOSe 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 JUNOSe 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
   ```

Specifying the PCMM Classifier Type

Use the following configuration statement to specify which version of the PCMM classifiers you are using:

```
shared sae configuration driver pcmm {
    disable-pcmm-i03-policy disable-pcmm-i03-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-i03-policy disable-pcmm-i03-policy
   ```

Specifying Port Access for Traffic Classification

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 JUNOSe 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**.

Creating a Classify-Traffic Condition

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";
   ```

Configuring Source Networks

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;
   ```

Configuring Source Grouped Networks

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;
   ```

Configuring Destination Networks

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;
   ```

Configuring Destination Grouped Networks

You can configure destination networks in grouped format. For JUNOS ASP and JUNOSe 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 JUNOSe 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;
   ```

Configuring Protocol Conditions

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* on page 226.
- If your condition consists of a protocol that is assigned with a parameter value, use the procedure in *Configuring Protocol Conditions with Parameters* on page 229.

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;
```

Configuring Protocol Conditions with Ports

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;
  }
}
```

Configuring Protocol Conditions with Parameters

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;
        }
      }
    }
```

Configuring TCP Conditions

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;
      }
    }
    ```

Configuring ICMP Conditions

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.

   [eedit 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;
   ```

Configuring IGMP Conditions

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;
```

Configuring IPSec Conditions

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.

   [eedit 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;
   ```

Configuring ToS Byte Conditions

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;
   ```

Configuring JUNOS Filter Conditions

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;
```

Configuring Application Protocol Conditions

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 source-port]
user@host# up

[edit policies group junos list staticnat rule nat traffic-condition ctc
application-protocol-condition apc proto-attr]
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;
    }
  }
}
```

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.

Configuring QoS Conditions

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";
   ```

Configuring Actions

Actions define the action taken on packets that match conditions in a policy rule. You create actions within policy rules. The type of action that you can create depends on the type of policy rule. See *Supported Conditions and Actions* on page 150.

Configure the action as described in the following sections:

- Configuring Color Actions on page 246
- Configuring DOCSIS Actions on page 247
- Configuring Filter Actions on page 251
- Configuring FlowSpec Actions on page 252
- Configuring Forward Actions on page 254
- Configuring Forwarding Class Actions on page 254
- Configuring GateSpec Actions on page 255

Configuring Color Actions

You can configure color actions for JUNOSe IPv6 policy rules.

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";
   ```

Configuring DOCSIS Actions

You can configure Data over Cable Service Interface Specifications (DOCSIS) actions for *PacketCable Multimedia Specification* (PCMM) policy rules.

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";
    ```

Configuring Filter Actions

Use this action to discard packets. You can configure filter actions for JUNOS filters and JUNOSe policy rules.

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";
```

Configuring FlowSpec Actions

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.

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";
```

Configuring Forward Actions

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 JUNOSe policy rules.

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";
   ```

Configuring Forwarding Class Actions

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.

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";
   ```

Configuring GateSpec Actions

You can configure GateSpec actions for PCMM policy rules. See *Session Class ID* on page 159 for more information.

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
```

Configuring Loss Priority Actions

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.

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";
   ```

Configuring Mark Actions

Use this action to mark packets. You can configure mark actions for JUNOSe and PCMM policy rules.

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
   ```

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";
   ```

Configuring NAT Actions

You can configure NAT actions for JUNOS ASP policy rules.

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;
}
```

Configuring Next-Hop Actions

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 JUNOSe policy rules.

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";
   ```

Configuring Next-Interface Actions

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 JUNOSe policy rules. On JUNOSe routers, you can use this action for both ingress and egress parts of the interface.

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 JUNOSe 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";
   ```

Configuring Next-Rule Actions

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.

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";
   ```

Configuring Policer Actions

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.

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;
   ```

Configuring the Packet Action for the Policer Action

The packet action specifies the action taken on a packet that exceeds its rate limits. You configure packet actions within policer actions.

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;
```

Configuring QoS Profile Attachment Actions

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 JUNOSe 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 *SRC-PE Solutions Guide, Chapter 1, Managing Tiered and Premium Services with QoS on JUNOSe Routers with the SRC CLI*.

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.

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 JUNOSe 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 JUNOSe 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";
```

Configuring Rate-Limit Actions

Use this action to define the quality of service. You can configure rate-limit actions for JUNOSe policy rules.

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;
    description description;
}

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;
}
```

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) 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
   ```

9. (Optional) Configure the rate-limit action for traffic flows that do not exceed the committed rate to one of the following:

   - Filter.

     ```
     [edit policies group junose list rate-limiter rule pr rate-limit rla]
     user@host# set committed-action filter
     ```

   - Forward.

     ```
     [edit policies group junose list rate-limiter rule pr rate-limit rla]
     user@host# set committed-action forward
     ```

   - 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
     ```

   - 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 mark mark-info]
     user@host# set committed-action parameter action action
     ```

10. (Optional) Configure the rate-limit action for traffic flows that exceed the committed rate but remain below the peak rate to one of the following:

    - Filter.

      ```
      [edit policies group junose list rate-limiter rule pr rate-limit rla]
      user@host# set conformed-action filter
      ```

    - Forward.

      ```
      [edit policies group junose list rate-limiter rule pr rate-limit rla]
      user@host# set conformed-action forward
      ```

- Mark. If you select mark, enter the mark values.

  ```
  [edit policies group junose list rate-limiter rule pr rate-limit rla]
  user@host# edit conformed-action mark mark-info

  [edit policies group junose list rate-limiter rule pr rate-limit rla
  conformed-action mark mark-info]
  user@host# set value value

  [edit policies group junose list rate-limiter rule pr rate-limit rla
  conformed-action mark mark-info]
  user@host# set mask mask
  ```

- 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
  conformed-action mark mark-info]
  user@host# set conformed-action parameter action action
  ```

11. (Optional) Configure the rate-limit action for traffic flows exceed the peak rate to one of the following:

    - Filter.

      ```
      [edit policies group junose list rate-limiter rule pr rate-limit rla]
      user@host# set exceed-action filter
      ```

    - Forward.

      ```
      [edit policies group junose list rate-limiter rule pr rate-limit rla]
      user@host# set exceed-action forward
      ```

    - Mark. If you select mark, enter the mark values.

      ```
      [edit policies group junose list rate-limiter rule pr rate-limit rla]
      user@host# edit exceed-action mark mark-info

      [edit policies group junose list rate-limiter rule pr rate-limit rla exceed-action
      mark mark-info]
      user@host# set value value

      [edit policies group junose list rate-limiter rule pr rate-limit rla exceed-action
      mark mark-info]
      user@host# set mask mask
      ```

    - 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 exceed-action
      mark mark-info]
      user@host# set exceed-action parameter action action
      ```

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;
    ```

Configuring Reject Actions

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.

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";
   ```

Configuring Routing Instance Actions

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.

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";
   ```

Configuring Scheduler Actions

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.

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";
   ```

Configuring Drop Profiles

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]";
   ```

Configuring Service Class Name Actions

You can configure service class name actions for PCMM policy rules. 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.";
   ```

Configuring Stateful Firewall Actions

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.

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";
```

Configuring Traffic-Class Actions

Use this action to put packets in a particular traffic class. You can configure traffic-class actions for JUNOSe policy rules.

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";
   ```

Configuring Traffic-Mirror Actions

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, you must 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 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.";
   ```

Configuring Traffic-Shape Actions

Traffic-shape actions specify the maximum rate of traffic transmitted on an interface. You can create traffic-shape actions in JUNOS shaping policy rules.

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";
   ```

Configuring User Packet Class Actions

Use this action to put packets in a particular user packet class. You can configure user packet class actions for JUNOSe IPv6 policy rules.

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 junose 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";
   ```

Chapter 11

Configuring and Managing Policies with Policy Editor

This chapter describes how to use Policy Editor to configure and manage policies. You can also use the following to configure and manage policies:

- To use the SRC CLI, see *Chapter 10, Configuring and Managing Policies with the SRC CLI*.
- To use the C-Web interface, see *SRC-PE C-Web Interface Configuration Guide, Chapter 21, Configuring and Managing Policies with the C-Web Interface*.

Topics in this chapter include:

Before You Configure 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 enter policy information into the Policy Editor fields, 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
3. Record the policy information about the worksheet.

Naming Objects

Object names must be unique and must conform to LDAP distinguished name (DN) constraints. You can provide your own object names, or Policy Editor can assist you by providing a name base for objects when you perform operations such as adding an object or copying and pasting an object into another folder.

You can configure whether or not Policy Editor suggests a name base for newly created objects and, if so, what Policy Editor uses as the name base. You can also specify whether or not object names in the navigation pane are prefixed with their object type. See Table 15 on page 168 in *Chapter 7, Using Policy Editor*.

If a name conflict occurs, the policy engine changes the object name by suffixing a number to the name separated by an underscore (_). The number is the next integer in sequence that does not cause a name conflict. You will see the new name in the navigation pane. For example, if policy group *internet-slow* exists in the organizational folder *XYZ*, then the policy engine assigns the name *internet-slow_1*. A subsequent policy group creation results in *internet-slow_2*.

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 fields can take expressions in addition to literal values. If you can enter an expression for a field, the expression type is noted in the field definition. For information about using and formatting expressions, see *Expressions* on page 404.

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 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, Policy Editor 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 a value greater than 100,000,000 is specified in Policy Editor, 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:

```
...
JunoseIpPolicyClaclRuleEntry ::= SEQUENCE {
...
junoseIpPolicyClaclRuleTosByte Integer32,
junoseIpPolicyClaclRuleTosMask 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.

Configuring Policy Folders

You use policy folders to organize policy groups. To create a policy folder:

1. In the Policy Editor navigation pane, right-click a Policy folder, and select **New > PolicyFolder**.

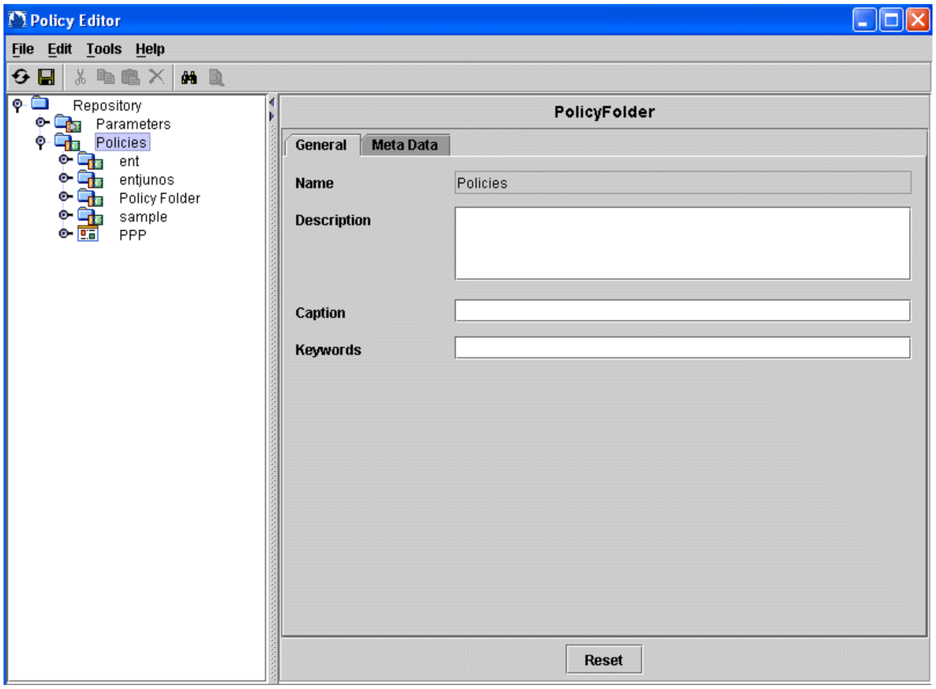
   The PolicyFolder Name dialog box appears.

2. Enter the policy folder name, and click **OK**.

   The new policy folder appears in the navigation pane.

3. Select the new policy folder.

   The PolicyFolder pane appears.



Name: Policies

4. Edit or accept the default values for the fields.

   See *Policy Folder Fields* on page 285.

5. Select **File > Save**.

Policy Folder Fields

In Policy Editor, you can modify the following fields in the PolicyFolder content pane.

Description

- Description of the policy folder.
- Value—Text
- Default—No value

Caption

- Short description of the policy folder.
- Value—Text
- Default—No value

Keywords

- Series of words that Policy Editor uses as a filter for keyword searches.
- Value—Text
- Default—No value

Configuring Policy Groups

You create policy groups within policy folders. To add a policy group:

1. In the Policy Editor navigation pane, right-click the **Policies** folder, and select **New > PolicyGroup**.

   or

   Cut or copy an existing policy group, and paste it to create a new policy group.

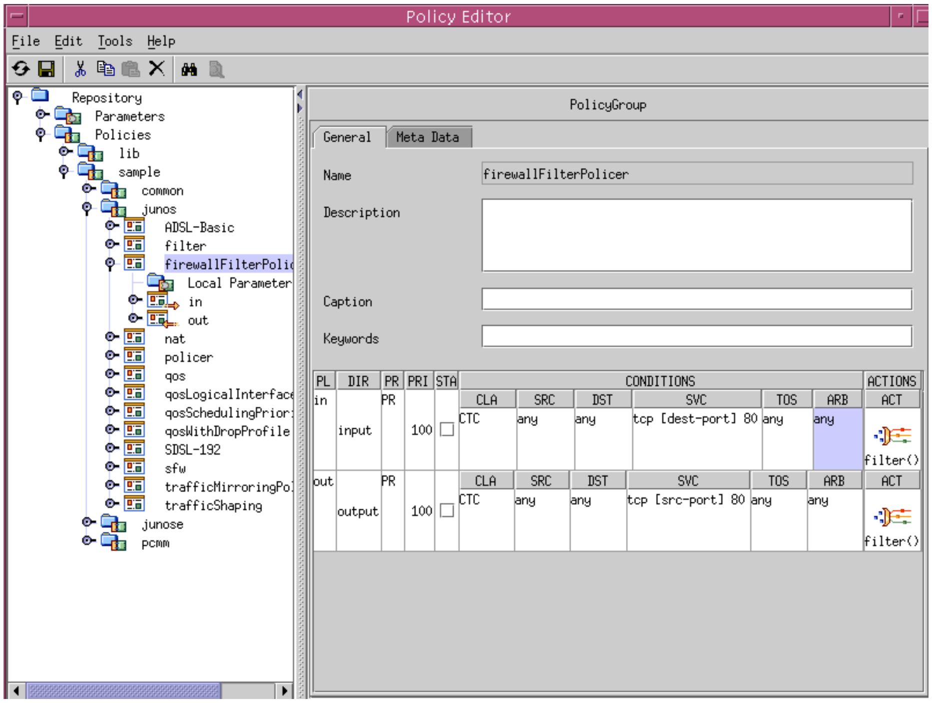
   The PolicyGroup Name dialog box appears.

2. Enter a unique name for the policy group, and click **OK**.

   The new policy group appears in the navigation pane.

3. Select the new policy group.

   The PolicyGroup pane appears.



Name: firewallFilterPolicer

Description:

Caption:

Keywords:

PL	DIR	PR	PRI	STA	CONDITIONS						ACTIONS
					CLA	SRC	DST	SVC	TOS	ARB	ACT
in	input	PR	100	□	CTC	any	any	tcp [dest-port] 80	any	any	filter()
out	output	PR	100	□	CTC	any	any	tcp [src-port] 80	any	any	filter()

4. Edit or accept the default values for the policy group fields.

   See *Policy Folder Fields* on page 285.

5. Select **File > Save**.

The PolicyGroup pane contains a table that summarizes the policy lists and rules that are within the policy group. See *Using the PolicyGroup Summary Table* on page 287.

Policy Group Fields

In Policy Editor, you can modify the following fields in a PolicyGroup content pane.

Description

- Description of the policy group.
- Value—Text
- Default—No value

Caption

- Short description of the policy group.
- Value—Text
- Default—No value

Keywords

- Series of words that Policy Editor uses as a filter for keyword searches.
- Value—Text
- Default—No value

Using the PolicyGroup Summary Table

The PolicyGroup pane contains a table that summarizes the policy lists and rules that are within the policy group. The fields in the table vary depending on the type of policy lists and rules in the policy group. Table 25 describes the fields in the policy group summary table.

Table 25: Fields in Policy Editor Summary Tables

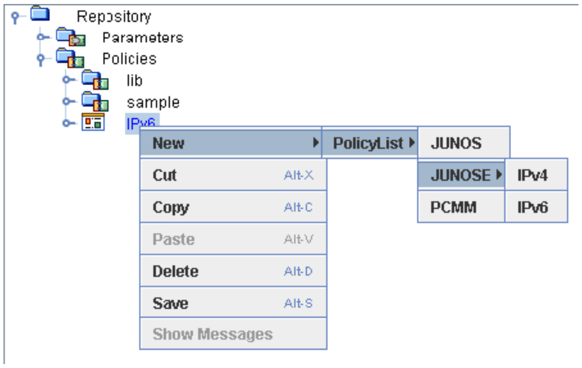
Field	Description	To Change the Field
ACT	Action applied to a policy group.	Double-click on an ACT cell. The corresponding action dialog box appears. or Right-click on an ACT cell. A pop-up menu appears from which you can edit or delete an action.
ARB	JUNOS filter conditions.	Double-click on an ARB cell. The Arbitrary Condition dialog box appears.
CLA	Classifier.	Double-click on a CLA cell. The condition pane appears.
DIR	Indicates the applicability of the policy list—whether the policy list associated with the policy group applies to egress or ingress traffic flow or both egress and ingress traffic flow. For PCMM policies, indicates whether the policy is for an upstream service flow or a downstream service flow.	Click on a DIR cell. A drop-down menu appears from which you can select a direction.
DST	Destination network matching.	Double-click on a DST cell. The Destination Network Condition dialog box appears.
FWC	Forwarding class QoS condition.	Double-click on an FWC cell. The QoS Condition dialog box appears.
PL	Name of the policy list.	Double-click on a PL cell. The PolicyList pane appears.
PR	Name of the policy rule.	Double-click on a PR cell. The PolicyRule pane appears.
PRI	Precedence that is applied to the actions of a policy rule.	Double-click on a PRI cell. A text cursor appears that allows you to type a new precedence.
ROLES	Indicates whether the policy list is a JUNOS policy list or a JUNOSe policy list.	Click on a ROLES cell. A drop-down menu appears from which you can select a role.
SRC	Source network matching.	Double-click on an SRC cell. The Source Network Condition dialog box appears.
STA	Indicates whether statistics accounting is enabled or disabled.	Click the check box. Checking the box enables accounting. Removing the check disables accounting.
SVC	Protocol and port matching.	Double-click on an SVC cell. The Protocol Condition dialog box appears.
TOS	Type of service matching.	Double-click on a TOS cell. The TOS Condition dialog box appears.

Configuring Policy Lists

When you add a policy list, you specify whether the policy list is for JUNOS routing platforms, JUNOSe routers, or a CMTS device (PCMM in Policy Editor). The type of policy list that you add controls the type of policy rules that you can add to the policy list.

To add a policy list:

1. In the Policy Editor navigation pane, right-click a policy group, and select **New > PolicyList**. Then select **JUNOS**, **JUNOSE**, or **PCMM**. If you select **JUNOSE**, then select either **IPv4** or **IPv6**.



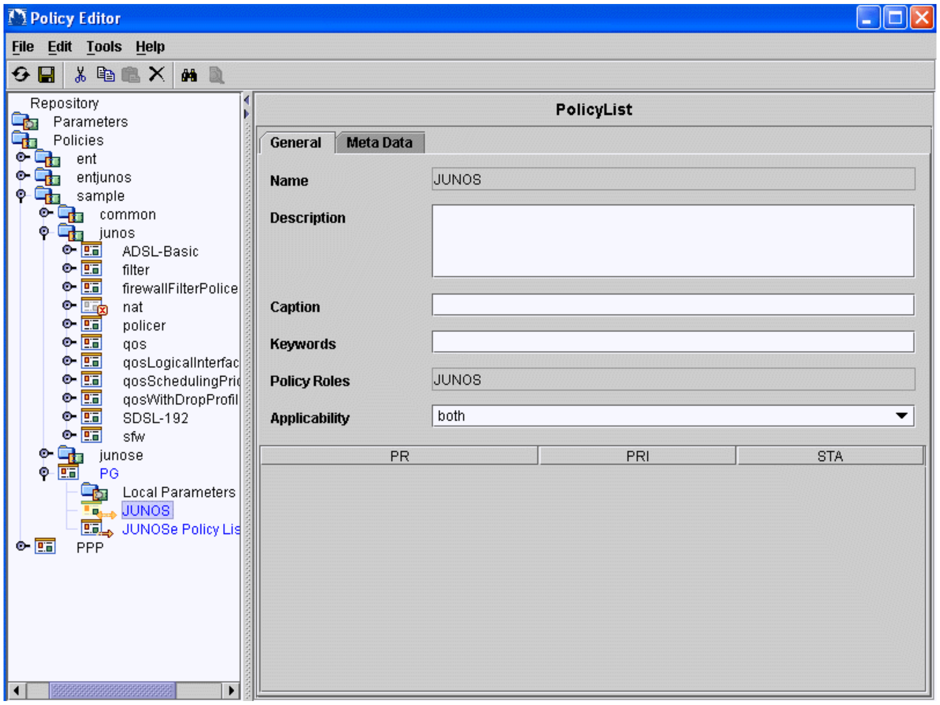
   The PolicyList Name dialog box appears.

2. Enter the policy list name, and click **OK**.

   The new policy list appears in the navigation pane.

3. Select the new policy list name.

   The PolicyList pane appears.



Name	JUNOS
Description	
Caption	
Keywords	
Policy Roles	JUNOS
Applicability	both

4. Edit or accept the default values for the policy list fields.

   See *Policy Folder Fields* on page 285.

5. Select **File** > **Save**.

Policy List Fields

In Policy Editor, you can modify the following fields in the PolicyList content pane.

Description

- Description of the policy list.
- Value—Text
- Default—No value

Caption

- Short description of the policy list.
- Value—Text
- Default—No value

Keywords

- Series of words that Policy Editor uses as a filter for keyword searches.
- Value—Text
- Default—No value

Policy Roles

- Indicates whether the policy list is a JUNOS policy list, a JUNOSe policy list (for IPv4 or IPv6), or a PCMM policy list. You cannot change this value.

Applicability

- Indicates 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 routing platforms, applicability determines the types of policy rules that you can create. For example, if you select both, you can create a JUNOS ASP or a JUNOS scheduler policy rule, but you cannot create a JUNOS filter.
- Value
  - input—Policy is applied to the input (ingress) side of the router interface. For PCMM policies, indicates that the policy is provisioned on upstream service flows (from the cable modem to the CMTS device).
  - output—Policy is applied to the output (egress) side of the router interface. For PCMM policies, indicates that the policy is provisioned on the downstream channel (from the CMTS device to the cable modem).
  - both—Policy is applied to both the input (ingress) and output (egress) side of the interface, or it is attached implicitly to the interface without indicating direction. *Both* is not valid for PCMM policies.

    In the case of JUNOS ASP policy rules, the policy is attached to both sides of the interface; for JUNOS scheduler policy rules, the policy is attached implicitly to the interface without indicating direction.
- Default
  - JUNOS policy lists—Both
  - JUNOSe IPv4 policy lists—Input
  - JUNOSe IPv6 policy lists—Input
  - PCMM policy lists—Input

Using the PolicyList Summary Table

The PolicyList pane contains a table that summarizes the policy rules that are within the policy list. It contains one row for each policy action that the policy list contains. The fields in the table vary depending on the type of policy rules that are contained in the policy list. You can modify policy rules from within the summary table, or you can modify them by selecting objects from the navigation pane. The fields in the summary table are explained in Table 25 on page 287.

Configuring 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 JUNOSe policy lists and PCMM policy lists. For JUNOS policy lists, you can create the following policy rule types:

- JUNOS ASP—Applicability of policy list must be both input and output.
- 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 AS 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.

Adding a Policy Rule

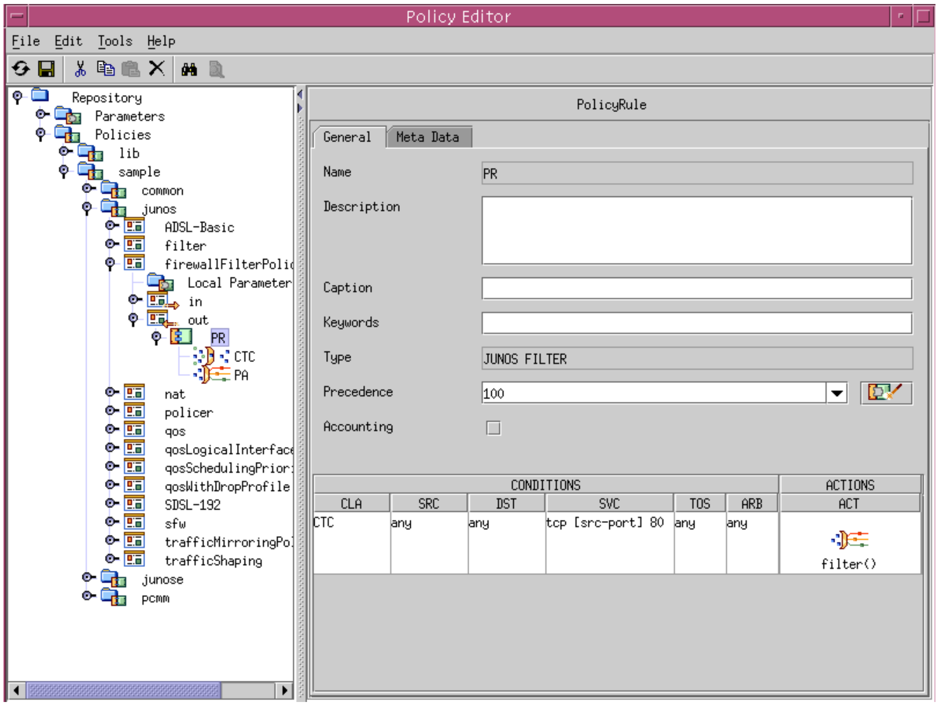
To add a policy rule:

1. In the navigation pane, right-click a policy list.
2. Select **New > PolicyRule**, and select a policy rule from the list.

   The PolicyRule Name dialog box appears.
3. Enter the Policy Rule name, and click **OK**.

   The new policy rule appears in the navigation pane.
4. Select the new Policy Rule object in the navigation pane.

   The PolicyRule pane appears.



Name PR

Description

Caption

Keywords

Type JUNOS FILTER

Precedence 100

Accounting □

CONDITIONS						ACTIONS
CLA	SRC	DST	SVC	TOS	ARB	ACT
CTC	any	any	tcp [src-port] 80	any	any	filter()

5. Edit or accept the default values for the policy rule fields.

   See *Policy Rule Fields* on page 293.

6. Select **File** > **Save**.

Policy Rule Fields

In Policy Editor, you can modify the following fields in the PolicyRule content pane.

Description

- Description of the policy rule.
- Value—Text
- Default—No value

Caption

- Short description of the policy rule.
- Value—Text
- Default—No value

Keywords

- Series of words that Policy Editor uses as a filter for keyword searches.
- Value—Text
- Default—No value

Precedence

- Precedence in which the policy rule is evaluated. Rules are evaluated from lowest to highest precedence value. Precedence is not a factor for JUNOS SCHEDULER and JUNOS POLICER policy rules. Precedence has meaning only if two rules have different classifiers and if those classifiers overlap. If this is the case and a packet is received that satisfies both classifiers, then only the action of the rule with the lower precedence value is performed. (See *Setting the Policy Rule Precedence* on page 292.)
- Value
  - For JUNOS and JUNOSe policies, integer in the range 0–32767
  - For PCMM policies, integer in the range 64–191
  - Parameter of type prPrecedence
- Default—100

Accounting

- Specifies whether accounting data is collected for the actions specified in the rule. (See *Collecting Accounting Statistics* on page 146.)
- Value—Checked or unchecked
- Default—Unchecked

Using the PolicyRule Summary Table

The PolicyRule pane contains a table that summarizes the conditions and actions that are within the policy rule. It contains one row for each action that the policy rule contains. The fields in the table vary depending on the type of conditions and actions that are contained in the policy rule. You can modify conditions and actions from within the summary table, or you can modify them by selecting objects from the navigation pane. The fields in the summary table are explained in Table 25 on page 287.

Configuring Classify-Traffic Conditions

You can create classify-traffic conditions in JUNOSe policy rules, in JUNOS ASP and JUNOS filter policy rules, and in PCMM policy rules. To create a classify-traffic condition:

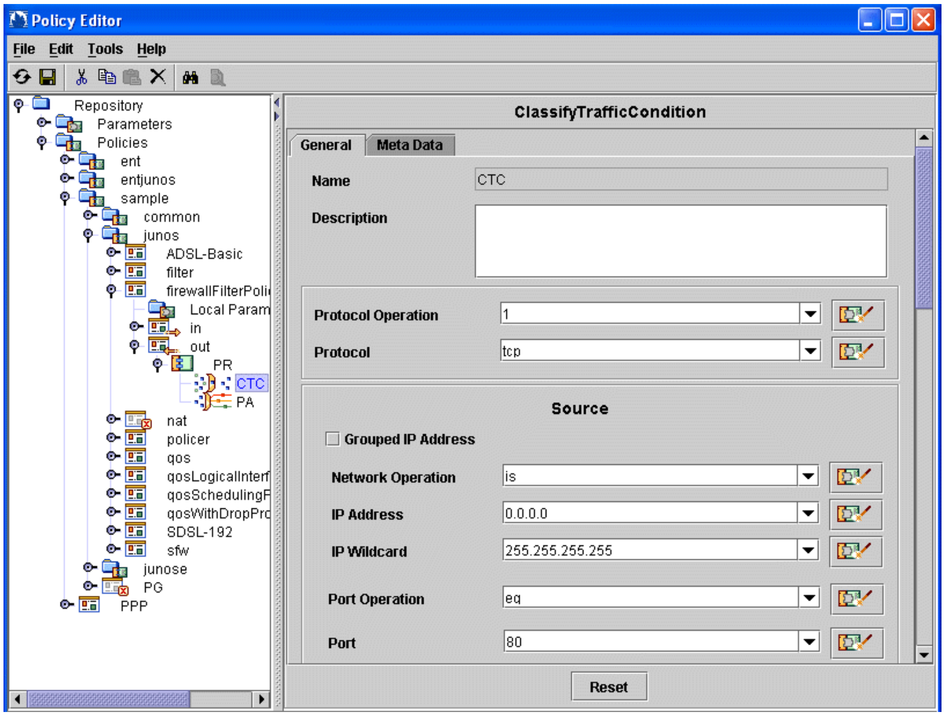
1. In the Policy Editor navigation pane, right-click a policy rule object, and select **New > Condition > ClassifyTrafficCondition**.

   The ClassifyTrafficCondition Name dialog box appears.

2. Enter a name, and click **OK**.

3. Select the new classify-traffic condition in the navigation pane.

   The new ClassifyTrafficCondition content pane appears.



Name: CTC

Description:

Protocol Operation: 1

Protocol: tcp

Source

□ Grouped IP Address

Network Operation: is

IP Address: 0.0.0.0

IP Wildcard: 255.255.255.255

Port Operation: eq

Port: 80

4. Edit or accept the default values for the classify-traffic condition fields.

   See *Classify-Traffic Condition Fields* on page 298.

   For information about configuring port ranges for traffic classifiers, see *Specifying Port Access for Traffic Classification* on page 296.

5. Select **File > Save**.

If you are configuring classifiers for PCMM policies, you can specify whether the classifier will be used in a PCMM I02 or I03 network. By default, the software translates classify-traffic conditions into PCMM I02 classifiers.

- See *Specifying the PCMM Classifier Type* on page 296.

For JUNOSe policies, you can specify that the SAE expands the classifier into multiple classifiers before it installs the policy on the router.

- See *Enabling Expansion of JUNOSe Classify-Traffic Conditions* on page 218.

Specifying the PCMM Classifier Type

To specify which version of the PCMM classifiers that you are using, configure the Router.pcmm.disableI03policy property in the SAE property file.

See *Modifying the SAE Property File* in *SRC-PE Subscribers and Subscriptions Guide, Chapter 5, Configuring Subscriber-Related Properties on the SAE on a Solaris Platform*.

For more information about PCMM classifiers, see *PCMM Classifiers* on page 160.

Router.pcmm.disableI03policy

- Specifies whether or not the SAE sends classifiers to the router that comply with PCMM 103.
- Value
  - true—The SAE sends classifiers that comply with PCMM I02 to the router.
  - false—The SAE sends classifiers that comply with PCMM I03 to the router.
- Guidelines—Set this property to false if your network deployment has CMTS devices that do not support PCMM I03.
- Default—true

Specifying Port Access for Traffic Classification

In the SRC software, the manner in which you specify a range of port numbers greater than or less than a specific value in a traffic classifier is different than the way you define a range in the configuration on JUNOSe routers.

In Policy Editor in the ClassifyTrafficCondition content pane, you specify ranges by setting values in the Port Operation field.

For information about accessing the configuration in the ClassifyTrafficCondition content pane, see *Configuring Classify-Traffic Conditions* on page 295.

For information about the Port Operation and Port fields, see *Source and Destination Network Fields* on page 300.

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. In the **Port Operation** field, enter **eq**.
2. In the **Port** field, enter the range of ports allowed.

   For example, to specify access to all port numbers greater than 10, specify 11..65535.

To configure port number greater than a define value by specifying which values are not allowed:

1. In the **Port Operation** field, enter **neq**.
2. In the **Port** field, 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. In the **Port Operatio**n field, enter **eq**.
2. In the **Port** field, 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. In the **Port Operation** field, enter **neq**.
2. In the **Port** field, enter the range of ports.

   For example, to specify access to all port numbers less than 10, specify 11..65535.

Classify-Traffic Condition Fields

In Policy Editor, you can modify the fields described in this section in the ClassifyTrafficCondition content pane.

The fields displayed in the ClassifyTrafficCondition pane change depending on the type of policy rule that holds the condition and on the type of protocol that you select in the Protocol field, as well as whether you select the Grouped IP Address and Raw check boxes. The classify-traffic condition fields are all described in the following sections:

- Direction Field on page 299
- Network Protocol Fields on page 299
- Source and Destination Network Fields on page 300
- Packet Length Field on page 303
- IP Protocol Fields on page 304
- ToS Byte on page 306
- TCP, ICMP, IGMP, and IPSec Protocol Fields on page 307
- JUNOS Filter Condition Fields on page 309
- Application Protocol Fields on page 311

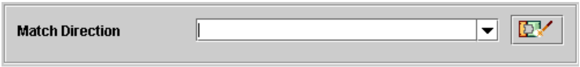
NOTE: PCMM classifiers support only the following fields:

- 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.

Direction Field

Appears only in JUNOS ASP policy rules.



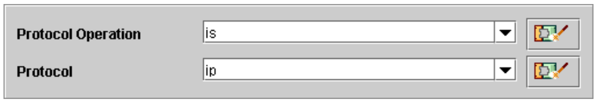
Match Direction

Match Direction

- Matches packets based on the direction of the packet flow. For stateful firewall actions, this value is used in place of the setting in the Applicability field of the policy list.
- Value
  - Predefined global parameter:
    - both—Valid only for stateful firewall actions
    - input
    - output
  - String expression
  - Parameter of type matchDirection
- Default—No value

Network Protocol Fields

This section of the pane specifies how protocols are matched.



Protocol Operation: is

Protocol: ip

Protocol Operation

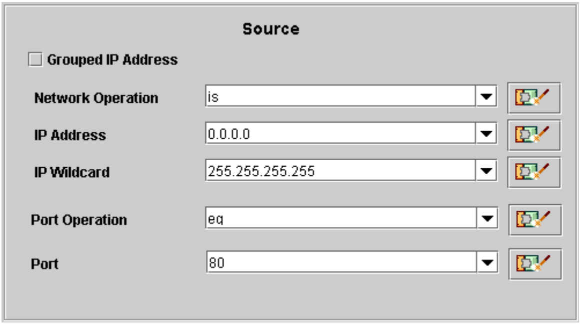
- Matches packets with the protocol that is either equal or not equal to the specified protocol.
- Value
  - Predefined global parameter:
    - is—Matches packets that are equal to the specified protocol
    - is_not—Matches any packets except those that are equal to the specified protocol
  - Boolean expression:
    - 1—is
    - 0—is_not
  - Parameter of type protocolOperation
- Default—1

Protocol

- Protocol matched by this classifier list.
- Value
  - Predefined global parameter—Select a protocol from the drop-down list
  - Protocol number in the range 0–257
  - For PCMM classifiers, there are two special protocol values:
    - 256 matches traffic that has any IP protocol value
    - 257 matches both TCP and UDP traffic
  - String expression
  - Parameter of type protocol

Source and Destination Network Fields

This section of the pane specifies source and destination networks. The Port Operation field appears only if you selected to match the TCP or UDP protocols. The Port field appears after you specify a port operation.



Source

☐ Grouped IP Address

Network Operation: is

IP Address: 0.0.0.0

IP Wildcard: 255.255.255.255

Port Operation: eq

Port: 80

Grouped IP Address

- If checked, the network operation, IP address, and IP wildcard attributes are grouped into one field called Network.
- For JUNOS ASP policy rules, you must check this box and enter IP addresses in prefix format; that is, IP address/prefix length.
- For JUNOSe IPv6 policy rules, you must check this box and enter IPv6 addresses with an IP mask in prefix length format. The network operation attribute is not supported.
- Value—Checked or unchecked
- Default—Unchecked

Network Operation

- Matches packets with an IP address that is either equal or not equal to the specified address and mask.
- Value
  - is—Matches the specified IP address and mask
  - not—Matches any IP address and mask except the specified address and mask
  - Parameter of type networkOperation
- Default—is

IP Address

- Number of the source or destination network or host.
- Value
  - IP address
  - Predefined global parameter:
    - gateway_ipAddress—IP address of the gateway as specified by the service object
    - interface_ipAddress—IP address of the router interface
    - service_ipAddress—IP address of the service as specified by the service object
    - user_ipAddress—IP address of the subscriber
    - virtual_ipAddress—Virtual portal address of the SSP that is used in redundant redirect server installations
  - Expression—For NAT actions, you can enter a range of addresses; for example, 10.10.13.1..10.10.13.100
  - Parameter of type address
- Default 0.0.0.0

IP Wildcard/IP Mask

- IP address mask applied to the IP address.
- Value
  - IP address mask
  - Predefined global parameter:
    - interface_ipMask—IP mask of the interface
    - service_ipMask—IP mask of the service as specified by the service object
    - user_ipMask—IP mask of the subscriber
  - Parameter of type addressMask
- Default—255.255.255.255

Network

- Network operation and IP subnets. This field appears only if the Grouped IP Address check box is checked.
- For JUNOS ASP policy rules, you must enter IP addresses in the format <address>/<prefix length>. The <address>/<mask> format is rejected by the router.
- For JUNOSe IPv6 policy rules, you must enter IPv6 addresses in the format <address>/<prefix length>. The network operation attribute is not supported.
- Value—Specify the subnet in one of the following formats:
  - [not] <address>/<mask> or <address>/<prefix length>
    - not is optional; include it to indicate that the condition matches every address that is not in the specified subnet
    - <address> and <mask> use dotted decimal notation
    - <prefix length> is a number in the range 0–32, and specifies how many of the first bits in the address specify the network
  - Expression—For example, pubIp/32

    where pubIp is a local address parameter and 32 is the prefix length
  - Parameter of type network
- Default—0.0.0.0/0.0.0.0

Port Operation

- Matches packets with a port that is either equal or not equal to the specified port.
- Value
  - Predefined global parameter:
    - eq—Matches packets that contain the specified port number
    - neq—Matches any packet except those that contain the specified port number
  - String
  - Parameter of type portOperation
- Guidelines—You can specify a range of port numbers as eq or neq to effectively specify a range greater than a specific value, or less than a specific value. For example to specify a port range greater than 49, you can specify eq for the port range 49..65536 or neq for the range 1..48.
- Default—No value

Port

- Source or destination ports.
- Value
  - Predefined global parameter:
    - service_port—Port of the service as specified by the service object
  - Integer in the range 0–65535
  - Expression—A range of port numbers; for example 10..20.

    Use a range of ports to specify port numbers that are greater than or less than a specified port number. For example:

    - To set a range of ports that is greater than 10, use 11..65535.
    - To set a range of ports that is less than 200, use 0..199.

    Note that PCMM 102 classifiers do not support port ranges. PCMM I03 classifiers do support port ranges.
  - Parameter of type port
- Guidelines—PCMM I02 does not support port ranges. If you are using PCMM 102 and you enter a range of port numbers, the software cannot translate the port, and it throws an exception.
- Default—No value

Packet Length Field

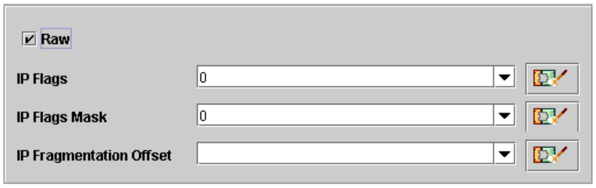
Matches packets according to packet length. This field appears only in JUNOS policy rules.

Packet Length (bytes)

- Matches on length of the packet. The length refers only to the IP packet, including the packet header, and does not include any layer 2 encapsulation overhead.
- Value
  - Number of bytes; all positive numbers and 0 are valid
  - Parameter of type packetLength
- Default—No value

IP Protocol Fields

In this section of the screen, you can configure values to match fields in the IP header.

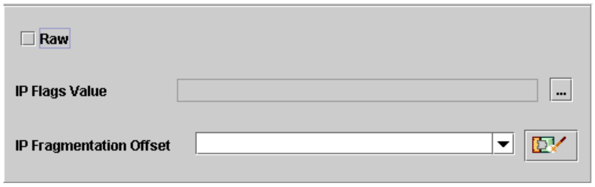


☑ Raw

IP Flags: 0

IP Flags Mask: 0

IP Fragmentation Offset:



☐ Raw

IP Flags Value:

IP Fragmentation Offset:

Raw

- Changes the view of the IP Flags section of the screen. You can configure IP flags and masks by number or by selecting values in a dialog box.
- Value—Checked or unchecked
- Default—Checked

IP Flags

- Value of the IP flags field in the IP header.
- Value
  - 0—Reserved
  - 1—Don't-fragment
  - 2—More fragments
  - Numeric expression
  - Parameter of type ipFlags
- Default—0

IP Flags Mask

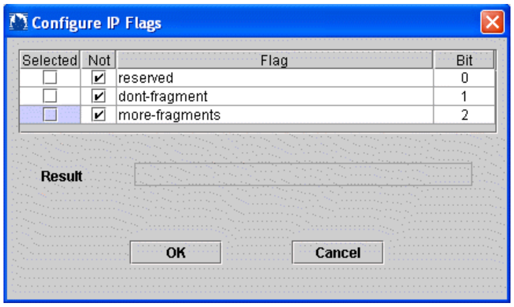
- Mask that is associated with the IP flag.
- Value
  - Integer in the range 0–7
  - Numeric expression
  - Parameter of type ipFlagsMask
- Default—0

IP Fragmentation Offset

- Value of the fragment offset field.
- Value
  - For JUNOSe routers:
    - eq 0—Equal to 0
    - eq 1—Equal to 1
    - gt 1—Greater than 1
    - any—Any value
  - For JUNOS routing platforms, integer in the range 0–8191
  - Numeric expression
  - Parameter of type fragOffset
- Default—No value

IP Flags Value

- If you deselect the Raw check box, Policy Editor displays the IP Flags Value field. Click ... next to the field to configure an IP flag. The Configure IP Flags dialog box appears.



Configure IP Flags

Selected	Not	Flag	Bit
☐	☑	reserved	0
☐	☑	dont-fragment	1
☐	☑	more-fragments	2

Result

To configure the IP flags:

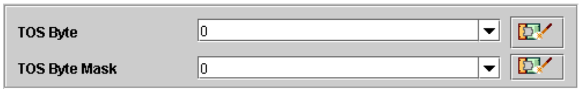
1. In the Selected column, select the IP flags that you want as part of the result string.
2. In the Not column, select the Not operator(s) that you want applied to the corresponding flag in the result string.

   You cannot check boxes in the Not column unless the check box in the corresponding Selected column is checked.
3. Click **OK**.

ToS Byte

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.



TOS Byte: 0

TOS Byte Mask: 0

TOS Byte

- Matches the value of the ToS byte in the IP packet header.
- Value
  - Integer in the range 0–255; uses whole 8 bits of the ToS byte
  - Numeric expression
  - Parameter of type tosByte
- Default—0

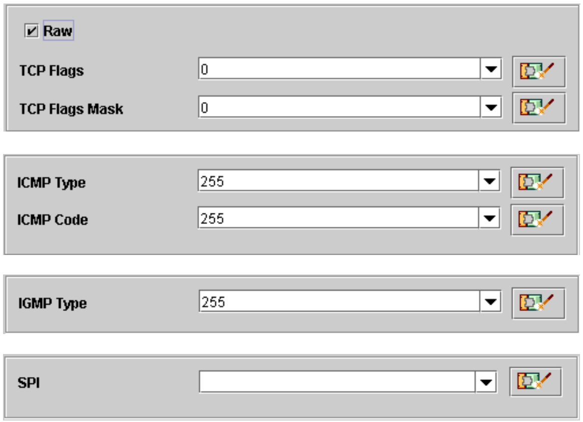
TOS Byte Mask

- Mask associated with the ToS byte.
- Value
  - Integer—Valid values are 0, 224, 252, 255
  - Numeric expression
  - Parameter of type tosByteMask
- Default—0

TCP, ICMP, IGMP, and IPSec Protocol Fields

If you specified the TCP, ICMP, IGMP, or the AH or ESP IPSec protocols, you can also specify the corresponding condition as shown in Figure 30.

Figure 30: Classify Conditions for TCP, ICMP, IGMP, and IPSec Protocols



☑ Raw

TCP Flags: 0

TCP Flags Mask: 0

ICMP Type: 255

ICMP Code: 255

IGMP Type: 255

SPI:

Raw

- Changes the view of the TCP section of the pane. You can configure TCP flags and masks by number or by selecting values in a dialog box.
- Value—Checked or unchecked
- Default—Checked

TCP Flags

- Value of the TCP flags field in the IP header.
- Value
  - Integer in the range 0–63
  - Numeric expression
  - Parameter of type tcpFlags
- Default—0

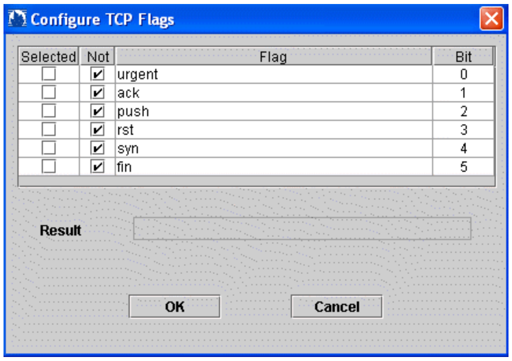
TCP Flags Mask

- Mask associated with TCP flags.
- Value
  - Integer in the range 0–63
  - Numeric expression
  - Parameter of type tcpFlagsMask
- Default—0

TCP Flags Value

- If you deselect the Raw check box, Policy Editor displays the IP Flags Value field. Click ... next to the field to configure a TCP flag.

  The Configure TCP Flags dialog box appears.



Selected	Not	Flag	Bit
□	☑	urgent	0
□	☑	ack	1
□	☑	push	2
□	☑	rst	3
□	☑	syn	4
□	☑	fin	5

To configure the TCP flags:

1. In the Selected column, select the TCP flags that you want as part of the result string.

2. In the Not column, select the Not operator(s) that you want applied to the corresponding flag in the result string.

   You cannot check boxes in the Not column unless the check box in the corresponding Selected column is checked.

3. Click **OK**.

ICMP Type

- Matches Internet Control Message Protocol (ICMP) packet type.
- Value
  - Integer in the range 0–255 that represents an ICMP packet type supported on the router or CMTS device
  - Numeric expression
  - Parameter of type icmpType
- Default—255

ICMP Code

- Matches ICMP code.
- Value
  - Integer in the range 0–255 that represents an ICMP code supported on the router or CMTS device
  - Numeric expression
  - Parameter of type icmpCode
- Default—255

IGMP Type

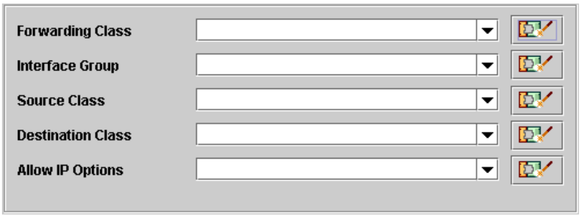
- IGMP packets that can be filtered by IGMP packet type or message name.
- Value
  - Integer in the range 0–255
  - Numeric expression
  - Parameter of type igmpType
- Default—255

SPI

- For IPSec classifiers, specifies the authentication header (AH) or the encapsulating security payload (ESP) security parameter index (SPI). This field appears only in JUNOS policy rules.
- Value
  - Integer in the range 0–255
  - Parameter of type ipSecSpi
- Default—No value

JUNOS Filter Condition Fields

The conditions described in this section appear only in JUNOS filter policy rules.



Forwarding Class

- Matches packets based on the name of a forwarding class.
- Value
  - String expression that matches a forwarding class on the router; for example, “assured-forwarding,” “best-effort,” “expedited-forwarding,” or “network-control”
  - Parameter of type forwardingClass
- Default—No value

Interface Group

- Matches packets based on the interface group on which the packet was received.
- Value
  - Integer in the range 0–4294967295
  - Numeric expression
  - Parameter of type interfaceGroup
- Default—No value

Source Class

- Matches packets based on source class. A source class is a set of source prefixes grouped together and given a class name. You would usually match source and destination classes for output firewall filters.
- Note that you cannot match on both source class and destination class at the same time. You must choose one or the other.
- Value
  - String expression that matches a source class that is configured on the router; for example, “gold-class”
  - Parameter of type trafficClassSpec
- Default—No value

Destination Class

- Matches packets based on destination class. A destination class is a set of destination prefixes grouped together and given a class name. You would usually match source and destination classes for output firewall filters.
- Note that you cannot match on both source class and destination class at the same time. You must choose one or the other.
- Value
  - String expression that matches a destination class that is configured on the router; for example, “gold-class”
  - Parameter of type trafficClassSpec
- Default—No value

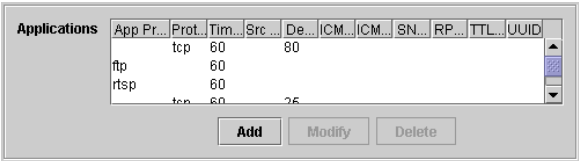
Allow IP Options

- Matches on IP options.
- Value
  - Numeric value of the IP option
  - String expression that matches a text synonym of an IP option on the router; for example, “loose-source-route,” “record-route,” “router-alert,” “strict-source-route,” or “timestamp”
  - Parameter of type allowIpOptions
- Default—No value

Application Protocol Fields

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.

The ClassifyTrafficCondition pane displays a table with configured application protocol conditions.



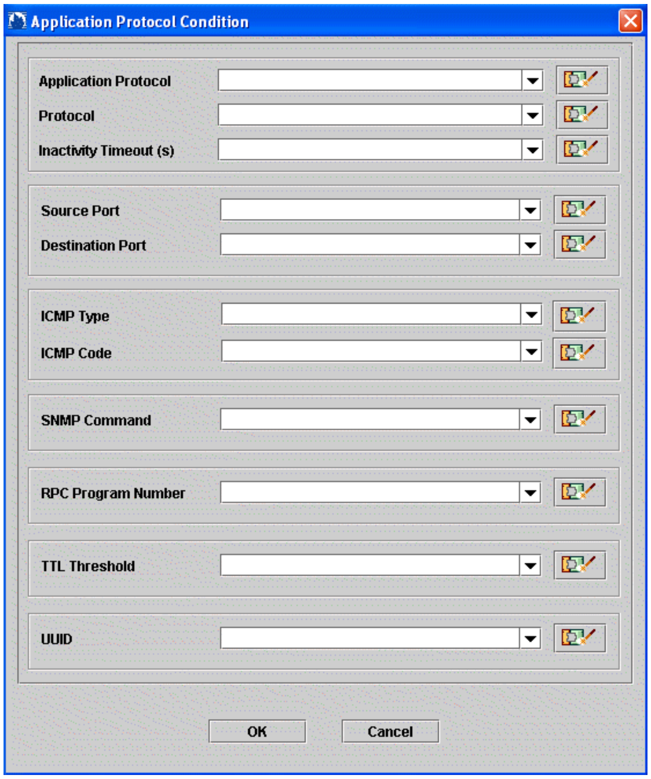
App Pr...	Prot...	Tim...	Src ...	De...	ICM...	ICM...	SN...	RP...	TTL...	UUID
	tcp	60		80						
ftp		60								
rtsp		60								
	tcp	60		25						

Configure the table as follows:

- To add an application protocol condition, click **Add**. Policy Editor displays the Application Protocol Condition dialog box.
- To modify a condition, select the condition, and click **Modify**. Policy Editor displays the Application Protocol Condition dialog box.
- To delete a condition, select the condition, and click **Delete**.

The Application Protocol Condition dialog box changes depending on the application protocol and protocol conditions that you select. Figure 31 shows an example of the dialog box with all possible fields.

Figure 31: Application Protocol Condition Dialog Box



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 field—the Application Protocol field. 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 field. “

Another map {applicationType = “tcp”, inactivityTimeout = 60, destinationPort = 80} supplies the protocol, inactivity timeout, and destination port.

You can enter the map expressions in the Application Protocol field.

You can also create a local parameter, add a map expression as the default value of the parameter, and then select the local parameter in the Application Protocol field.

Filling in Application Protocol Fields

This section describes the fields in the Application Protocol Condition dialog box.

Application Protocol

- Application protocol to match.
- Value
  - Predefined global parameter—Select a protocol from the pull-down list
  - String expression that matches an application protocol name supported on the router
  - Map expression—See *Using Map Expressions in Application Protocol Conditions* on page 312
  - Parameter of type applicationProtocol
- Default—No value

Protocol

- Network protocol to match.
- Value
  - Predefined global parameter—Select a protocol from the drop-down list
  - Integer in the range 0–255
  - Numeric expression
  - Parameter of type protocol
- Default—No value

Inactivity Timeout (s)

- Length of time the application is inactive before it times out.
- Value
  - Number of seconds in the range 4–65535
  - Numeric expression
  - Parameter of type timeout
- Default—Unspecified; the router’s default value is used

Source Port

- TCP or UDP source port.
- Value
  - Predefined parameter:
    - service_port—Service port as specified by the service object
  - Integer in the range 0–65535
  - String expression that matches a port name supported on the router; for example, “http”
  - Parameter of type port
- Default—No value

Destination Port

- TCP or UDP destination port.
- Value
  - Predefined parameter:
    - service_port—Service port as specified by the service object
  - Integer in the range 0–65535
  - String expression that matches a port name or number supported on the router; for example, “http”
  - Parameter of type port
- Default—No value

ICMP Type

- ICMP packet type.
- Value
  - Integer in the range 0–255 that represents an ICMP packet type supported on the router
  - Numeric expression
  - Parameter of type icmpType
- Default—No value

ICMP Code

- ICMP code.
- Value
  - Integer in the range 0–255 that represents an ICMP code supported on the router
  - Numeric expression
  - Parameter of type icmpCode
- Default—No value

SNMP Command

- SNMP command for packet matching.
- Value
  - Predefined parameter:
    - get
    - get_next
    - set
    - trap
  - String expression that matches an SNMP command supported on the router
  - Parameter of type snmpCommand
- Default—No value

RPC Program Number

- For the remote procedure call (RPC) application protocol, specifies an RPC program number.
- Value
  - Integer—RPC or DCE program number in the range 100000–400000
  - Numeric expression
  - Parameter of type rpcProgramNumber
- Default—No value

TTL Threshold

- For the traceroute application protocol, specifies the traceroute time-to-live (TTL) threshold value. This value sets the acceptable level of network penetration for trace routing.
- Value
  - Integer in the range 0–255
  - Numeric expression
  - Parameter of type traceRouteTtlThreshold
- Default—No value

UUID

- For the DCE RPC application protocol, specifies the universal unique identifier (UUID).

  For information about UUIDs, see http://www.opengroup.org/onlinepubs/9629399/apdxa.htm.
- Value
  - Hexadecimal value
  - Numeric expression
  - Parameter of type dceRpcUuid
- Default—dceRpcUuid

Configuring QoS Conditions

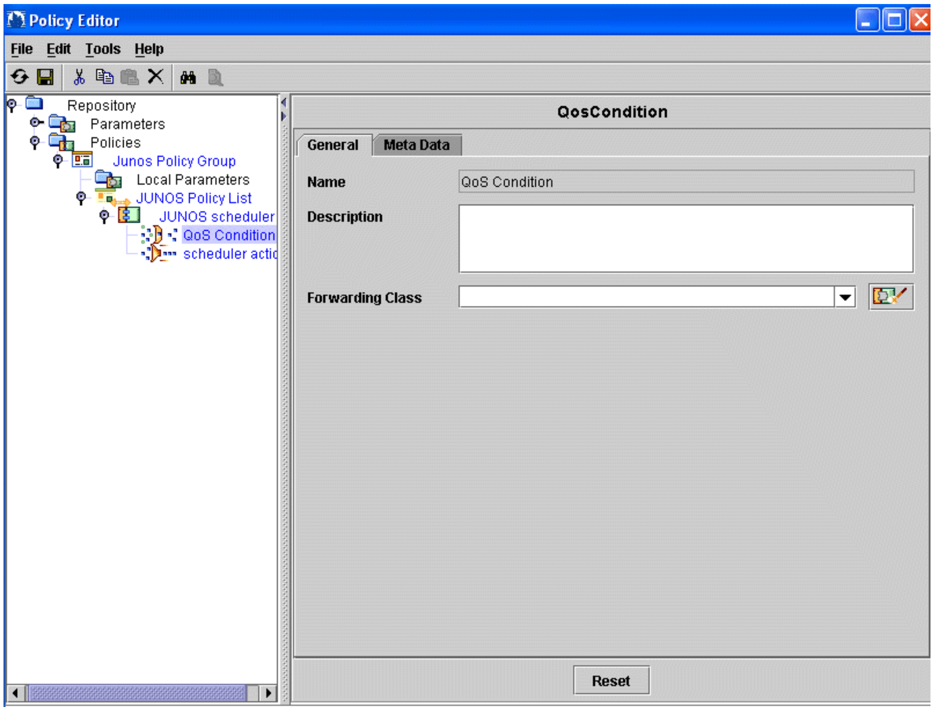
You can create QoS conditions within JUNOS scheduler policy rules. To create a QoS condition:

1. In the navigation pane, right-click a JUNOS scheduler policy rule object, and select **New > Condition > QosCondition**.

   The QosConditon Name dialog box appears.
2. Enter a name, and click **OK**.

3. Select the new QoS condition in the navigation pane.

   The QosConditon pane appears.



Name: QoS Condition

4. Edit or accept the default values for the QoS condition fields.

   See *QoS Condition Fields* on page 317.

5. Select **File** > **Save**.

QoS Condition Fields

In Policy Editor, you can modify the following fields in the QoSCondition content pane.

Description

- Description of the QoS condition.
- Value—Text
- Default—No value

Forwarding Class

- Matches packets based on forwarding class.
- Value
  - String expression that matches forwarding classes that are configured on the router; for example, “assured-forwarding,” “best-effort,” “expedited-forwarding,” or “network-control”
  - Parameter of type forwardingClass
- Default—No value

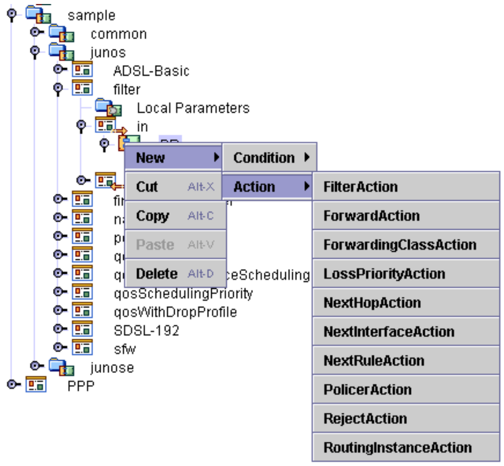
Configuring Actions

Actions define the action taken on packets that match conditions in a policy rule. You create actions within policy rules. The type of action that you can create depends on the type of policy rule. See *Supported Conditions and Actions* on page 150.

Adding Actions

To add an action:

1. In the navigation pane, right-click a policy rule.
2. Click **New > Action,** and select an action from the list.



The <Action> Name dialog box appears.

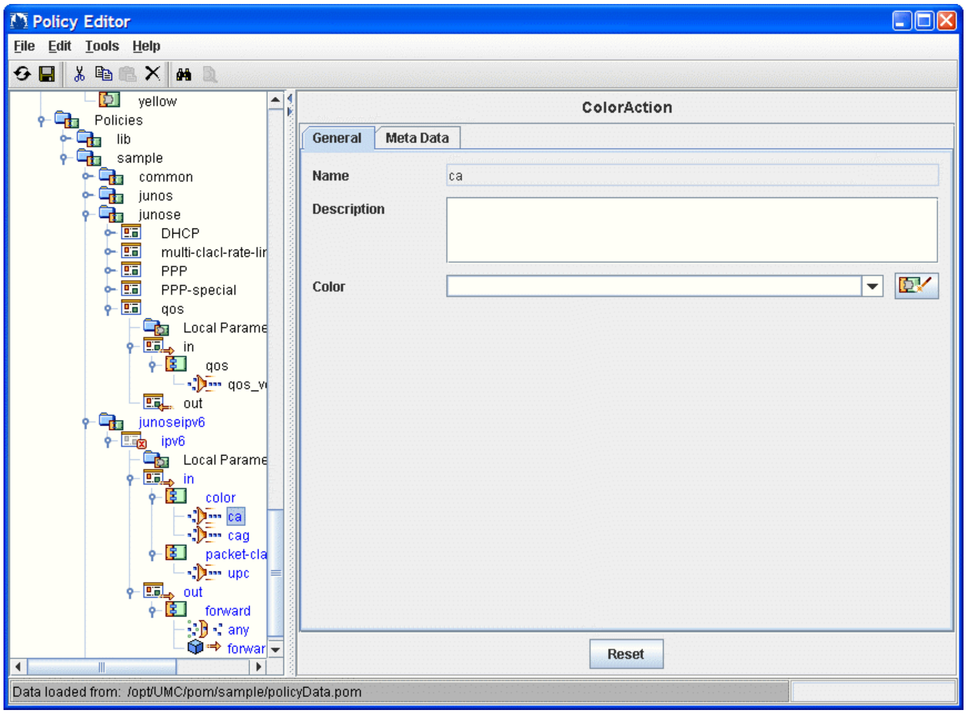
3. Enter a name, and click **OK**.
4. Select the new action in the navigation pane.

5. Configure the action as described in the following sections:

   - Configuring Color Actions on page 320
   - Configuring DOCSIS Actions on page 321
   - Configuring Filter Actions on page 325
   - Configuring FlowSpec Actions on page 326
   - Configuring Forward Actions on page 329
   - Configuring Forwarding Class Actions on page 330
   - Configuring GateSpec Actions on page 331
   - Configuring Loss Priority Actions on page 333
   - Configuring Mark Actions on page 334
   - Configuring NAT Actions on page 335
   - Configuring Next-Hop Actions on page 337
   - Configuring Next-Interface Actions on page 339
   - Configuring Next-Rule Actions on page 341
   - Configuring Policer Actions on page 342
   - Configuring QoS Profile Attachment Actions on page 344
   - Configuring Rate-Limit Actions on page 346
   - Configuring Reject Actions on page 350
   - Configuring Routing Instance Actions on page 351
   - Configuring Scheduler Actions on page 352
   - Configuring Service Class Name Actions on page 358
   - Configuring Stateful Firewall Actions on page 359
   - Configuring Traffic-Class Actions on page 360
   - Configuring Traffic-Mirror Actions on page 361
   - Configuring Traffic-Shape Actions on page 363
   - Configuring User Packet Class Actions on page 364

Configuring Color Actions

You can configure color actions for JUNOSe IPv6 policy rules.



Name: ca

Description:

Color:

Data loaded from: /opt/UMC/pom/sample/policyData.pom

Description

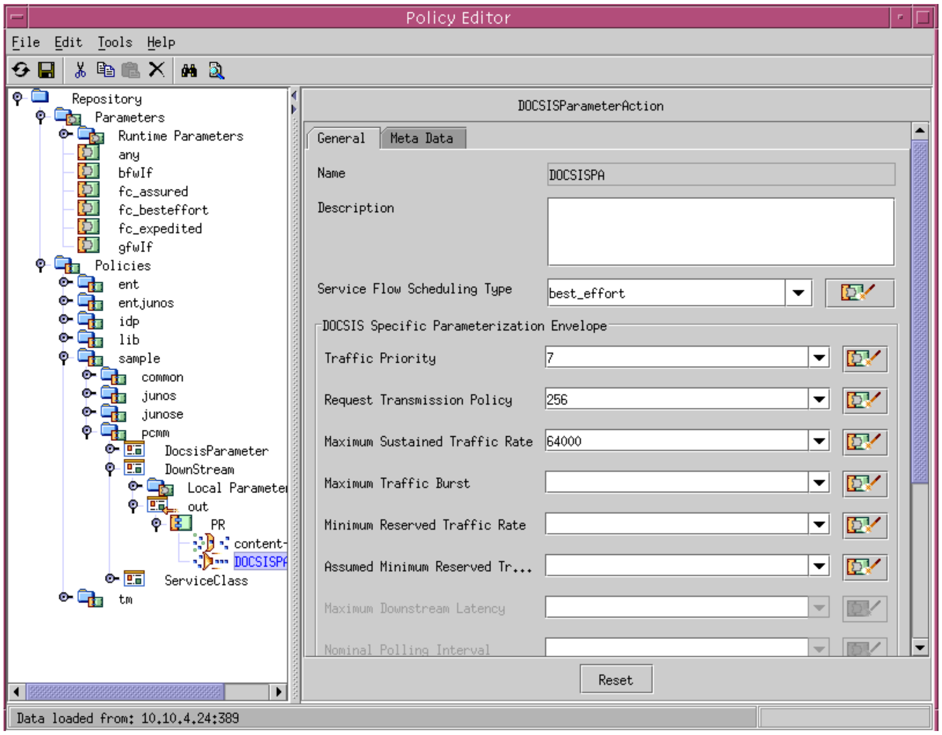
- Description of the action.
- Value—Text
- Default—No value

Color

- Color that is applied to a packet when it passes through the router.
- Value
  - Integer in the range 1–3
    - 1 is green; indicating a low drop preference
    - 2 is yellow; indicating a medium drop preference
    - 3 is red; indicating a high drop preference
  - Parameter of type color
- Default—No value

Configuring DOCSIS Actions

You can configure Data over Cable Service Interface Specifications (DOCSIS) actions for *PacketCable Multimedia Specification* (PCMM) policy rules.



Name: DOCSISPA

Description:

Service Flow Scheduling Type: best_effort

DOCSIS Specific Parameterization Envelope

Traffic Priority: 7

Request Transmission Policy: 256

Maximum Sustained Traffic Rate: 64000

Maximum Traffic Burst:

Minimum Reserved Traffic Rate:

Assumed Minimum Reserved Tr...:

Maximum Downstream Latency:

Nominal Polling Interval:

Data loaded from: 10.10.4.24:389

Service Flow Scheduling Type

- Scheduling types for service flows. The scheduling type that you select determines which fields are available in the DOCSIS action.
- Value
  - Predefined global parameter. For information about each DOCSIS service scheduling type, see Table 12 on page 157.
    - best_effort
    - unsolicited_grant
    - down_stream
    - unsolicited_grant_with_activity_detection
    - real_time
    - non_real_time
  - Parameter of type trafficProfileType
- Default—No value

Traffic Priority

- 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.
- Value
  - Number in the range 0–7, where 0 is the lowest priority and 7 is the highest priority
  - Parameter of type trafficPriority
- Default—No value

Request Transmission Policy

- Interval usage code that the cable modem uses for upstream transmission requests and packet transmissions for this service flow, and specifies whether requests can be piggybacked with data. Also, for data packets transmitted on this service flow, specifies whether packets can be concatenated, fragmented, or have their payload headers suppressed. For UGS service flows, this field also specifies how to treat packets that do not fit into the UGS grant.
- Value
  - 4-byte bit field; the valid range is 0–511
  - Parameter of type requestTransmissionPolicy
- Default—No value

Maximum Sustained Traffic Rate

- Maximum sustained rate at which traffic can operate over the service flow.
- Value
  - Predefined global parameter:
    - interface_speed—Speed of the subscriber’s DOCSIS interface
  - Number of bits per second in the range 0–4294967295
  - Numeric expression
  - Parameter of type rate
- Default—No value

Maximum Traffic Burst

- Maximum burst size for the service flow. This parameter has no effect unless you configure a nonzero value for the maximum traffic rate.
- Value
  - Predefined global parameter:
    - interface_speed—Speed of the subscriber’s DOCSIS interface
  - Number of bytes in the range 1522–4294967295
  - Numeric expression
  - Parameter of type burst
- Default—No value

Minimum Reserved Traffic Rate

- Guaranteed minimum rate that is reserved for the service flow.
- Value
  - Predefined global parameter:
    - interface_speed—Speed of the subscriber's DOCSIS interface
  - Number of bits per second in the range 0–4294967295; a value of 0 means that no bandwidth is reserved for the service flow
  - Numeric expression
  - Parameter of type rate
- Default—No value

Assumed Minimum Reserved Traffic Rate Packet Size

- 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 field.
- Value
  - Number of bytes in the range 0–65535
  - Numeric expression
  - Parameter of type packetLength
- Default—No value

Maximum Downstream Latency

- 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.
- Value
  - Number of microseconds in the range 0–4294967295
  - Numeric expression
  - Parameter of type maxLatency
- Default—No value

Nominal Polling Interval

- Nominal interval between successive unicast request opportunities for this service flow.
- Value
  - Number of microseconds in the range 0–4294967295
  - Numeric expression
  - Parameter of type interval
- Default—No value

Tolerated Poll Jitter

- Maximum amount of time that unicast request intervals can be delayed beyond the nominal polling interval. Delaying requests allows the service flow scheduler to fit as much data as possible in an upstream packet, thereby reducing fragmentation.
- Value
  - Number of microseconds in the range 0–4294967295
  - Numeric expression
  - Parameter of type jitter
- Default—No value

Unsolicited Grant Size

- Size of the individual data grants provided to the service flow.
- Value
  - Number of bytes in the range 0–65535
  - Numeric expression
  - Parameter of type grantSize
- Default—No value

Grants Per Interval

- Actual number of data grants given to the service flow during each nominal grant interval.
- Value
  - Integer in the range 0—127
  - Numeric expression
  - Parameter of type interval
- Default—No value

Nominal Grant Interval

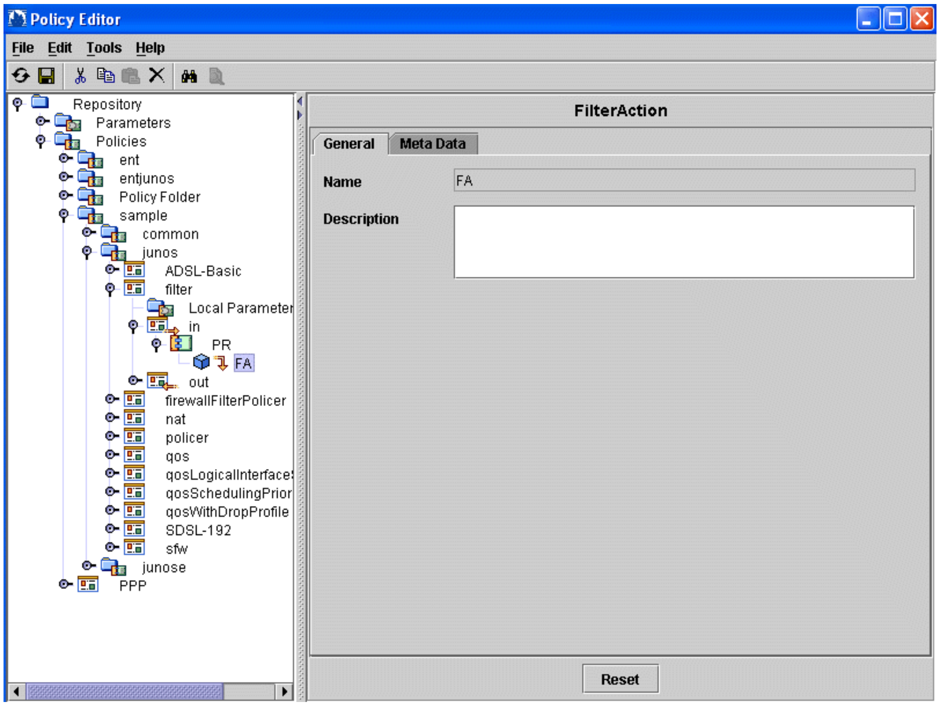
- Nominal interval between successive unsolicited data grant opportunities for this service flow.
- Value
  - Number of microseconds in the range 0–4294967295
  - Numeric expression
  - Parameter of type interval
- Default—No value

Tolerated Grant Jitter

- Maximum amount of time that the transmission opportunities can be delayed beyond the nominal grant interval.
- Value
  - Number of microseconds in the range 0–4294967295
  - Numeric expression
  - Parameter of type jitter
- Guidelines—A jitter buffer can stop latency, but an improperly sized buffer can cause additional latency.
- Default—No value

Configuring Filter Actions

Use this action to discard packets. You can configure filter actions for JUNOS filters and JUNOSe policy rules.



Name: FA

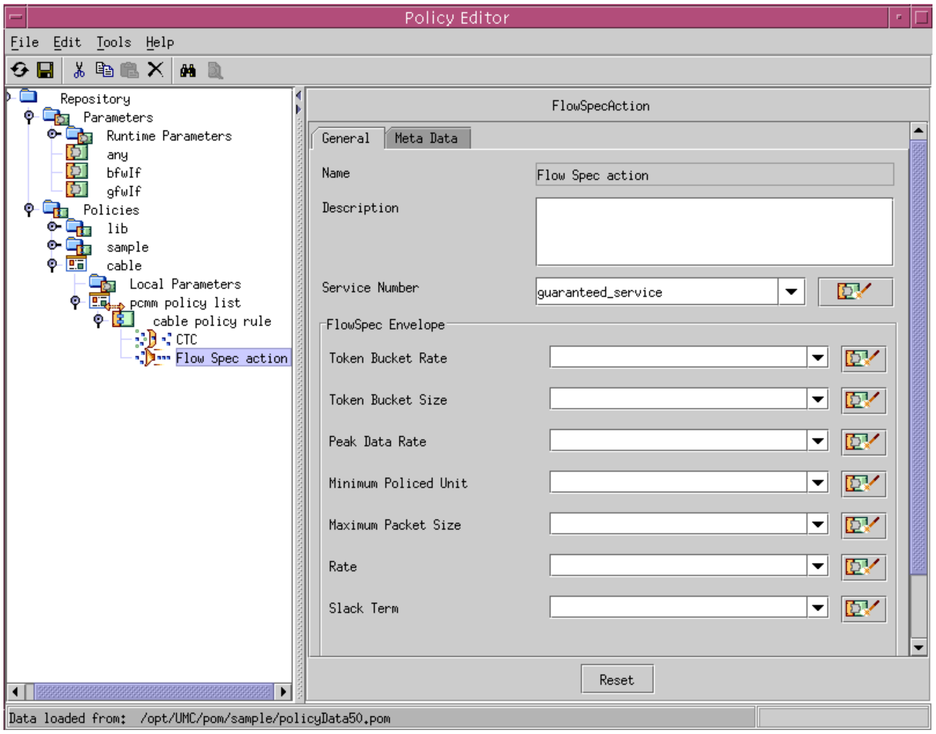
Description:

Description

- Description of the action.
- Value—Text
- Default—No value

Configuring FlowSpec Actions

You can configure FlowSpec actions for PCMM policy rules.



FlowSpecAction

Name: Flow Spec action

Description:

Service Number: guaranteed_service

Data loaded from: /opt/UMC/pom/sample/policyData50.pom

Service Number

- Type of FlowSpec service.
- Value
  - Predefined global parameter:
    - controlled_load_service—Provides minimum bandwidth guarantees, but not latency and delay guarantees. A controlled-load service can contain only traffic specification (TSpec) token-bucket parameters, and not service request specification (RSpec) parameters.
    - guaranteed_service—Provides both bandwidth and latency and delay guarantees. A guaranteed service can contain both TSpec and RSpec parameters.
  - Parameter of type serviceNumber
- Default—No value

Token Bucket Rate

- Guaranteed minimum rate that is reserved for the service flow. Token bucket rate is a TSpec parameter.
- Value
  - Predefined global parameter:
    - interface_speed—Speed of the subscriber's DOCSIS interface
  - Number of bits per second in the range 0–4294967295
  - Numeric expression
  - Parameter of type rate
- Default—No value

Token Bucket Size

- Maximum burst size for the service flow. Token bucket size is a TSpec parameter.
- Value
  - Number of bits per second in the range 1522–4294967295
  - Numeric expression
  - Parameter of type tokenBucketSize
- Guidelines—This parameter has no effect unless you configure a nonzero value for the maximum traffic rate.
- Default—No value

Peak Data Rate

- Amount of bandwidth over the committed rate that is allocated to accommodate excess traffic flow over the committed rate. Peak data rate is a TSpec parameter.
- Value
  - Predefined global parameter:
    - interface_speed—Speed of the subscriber's DOCSIS interface
  - Number of bits per second in the range 0–4294967295
  - Numeric expression
  - Parameter of type rate
- Default—No value

Minimum Policed Unit

- 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 field. Minimum policed unit is a TSpec parameter.
- Value
  - Number of bytes in the range 0–65535
  - Numeric expression
  - Parameter of type policedUnit
- Default—No value

Maximum Packet Size

- Maximum packet size for the FlowSpec. Maximum packet size is a TSpec parameter.
- Value
  - Number of bytes in the range 0–4294967295
  - Numeric expression
  - Parameter of type packetLength
- Default—No value

Rate

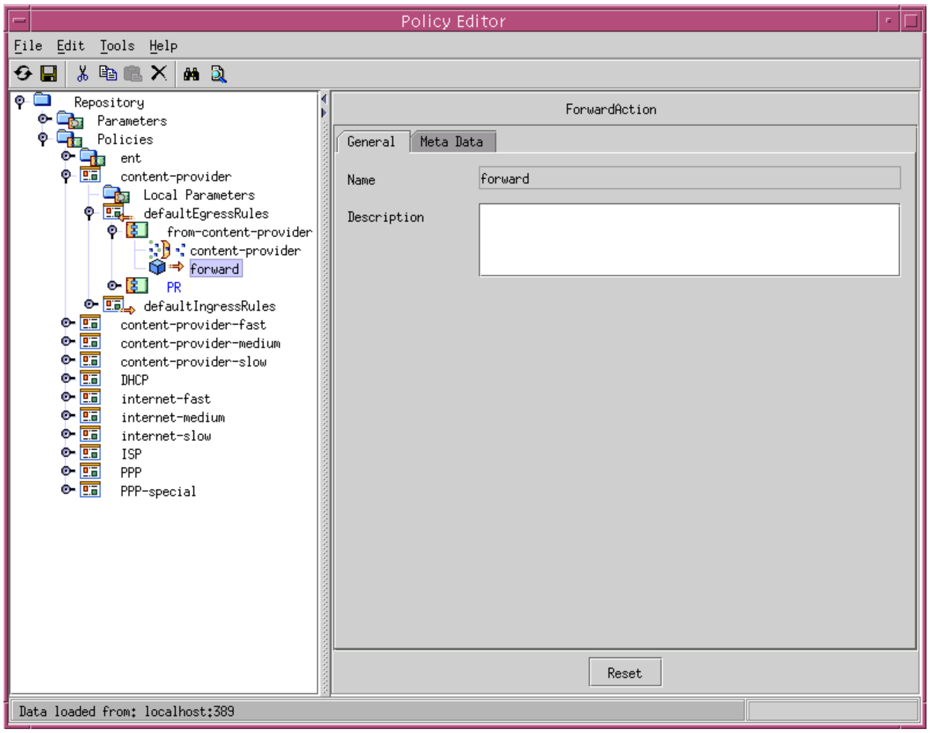
- Average rate. Rate is an RSpec parameter.
- Value
  - Predefined global parameter:
    - interface_speed—Speed of the subscriber's DOCSIS interface
  - Number of bits per second in the range 0–4294967295
  - Numeric expression
  - Parameter of type rate
- Default—No value

Slack Term

- Amount of slack in the bandwidth reservation that can be used without redefining the reservation. Slack is the difference between the desired delay and the actual delay obtained with the current bandwidth reservation. It allows some flexibility in bandwidth reservations. Slack term is an RSpec parameter.
- Value
  - Integer in the range 0–4294967295
  - Numeric expression
  - Parameter of type slackTerm
- Default—No value

Configuring Forward Actions

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 JUNOSe policy rules.



Name: forward

Description:

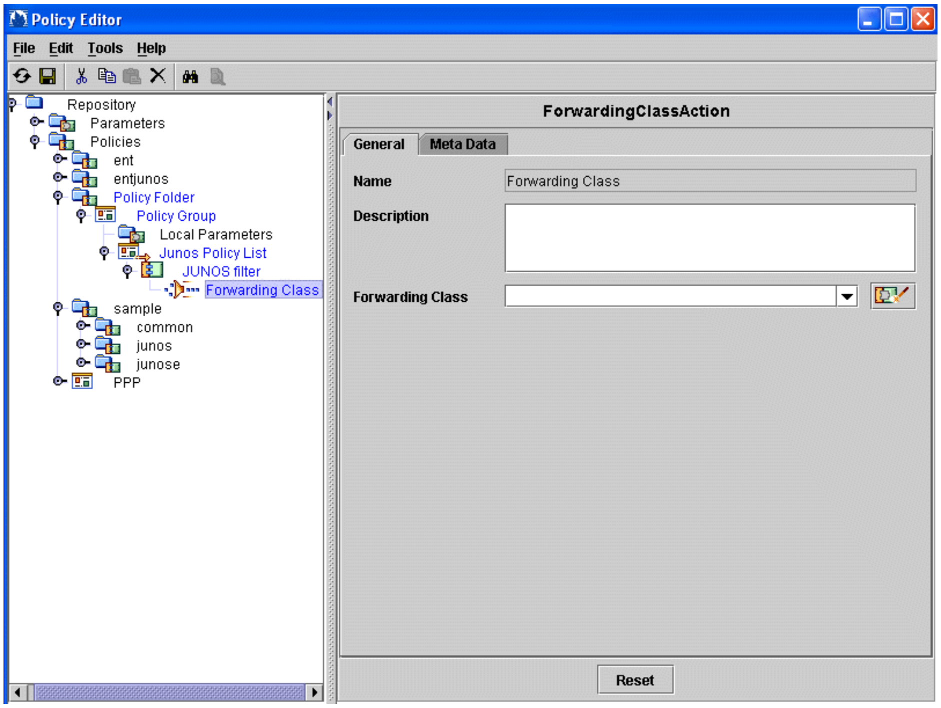
Data loaded from: localhost:389

Description

- Description of the action.
- Value—Text
- Default—No value

Configuring Forwarding Class Actions

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.



Name: Forwarding Class

Description:

Forwarding Class:

Description

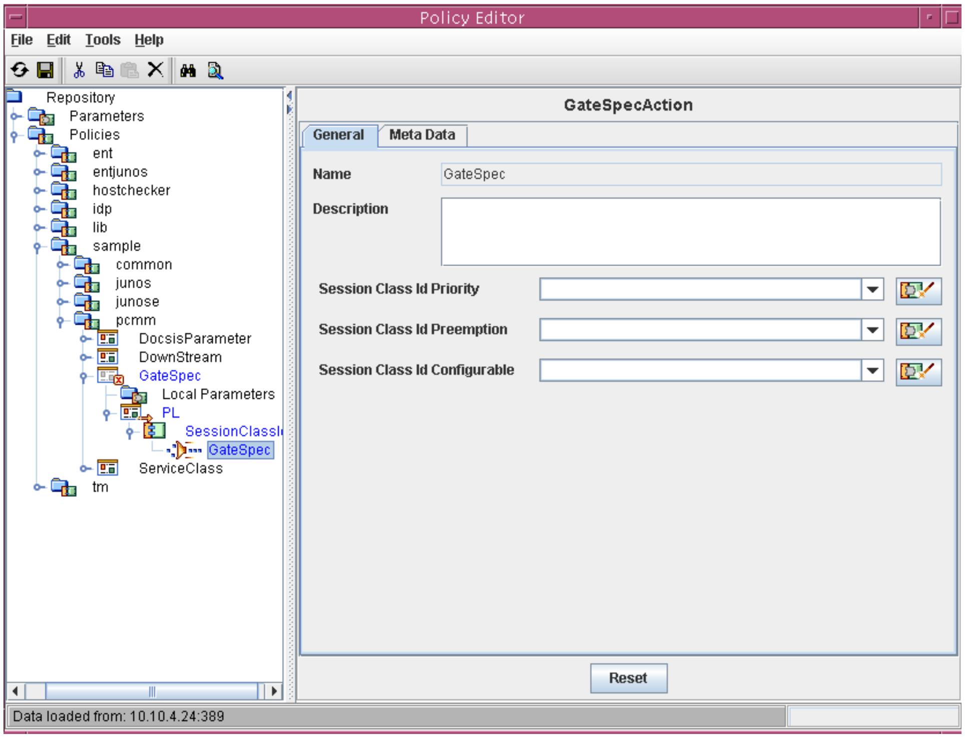
- Description of the action.
- Value—Text
- Default—No value

Forwarding Class

- Name of the forwarding class assigned to packets.
- Value
  - String expression that matches a forwarding class that is configured on the router; for example, “assured-forwarding,” “best-effort,” “expedited-forwarding,” or “network-control”
  - Parameter of type forwardingClass
- Default—No value

Configuring GateSpec Actions

You can configure GateSpec actions for PCMM policy rules. See *Session Class ID* on page 159 for more information.



Name: GateSpec

Data loaded from: 10.10.4.24:389

Description

- Description of the action.
- Value—Text
- Default—No value

Session Class Id Priority

- Priority bits in the session class ID. The priority field describes the relative importance of the session as compared with other sessions generated by the same policy decision point.
- Value
  - Number in the range 0–7, where 0 is low priority and 7 is high priority
  - String expression
  - Parameter of type sessionClassIdPriority
- Default—No value

Session Class Id Preemption

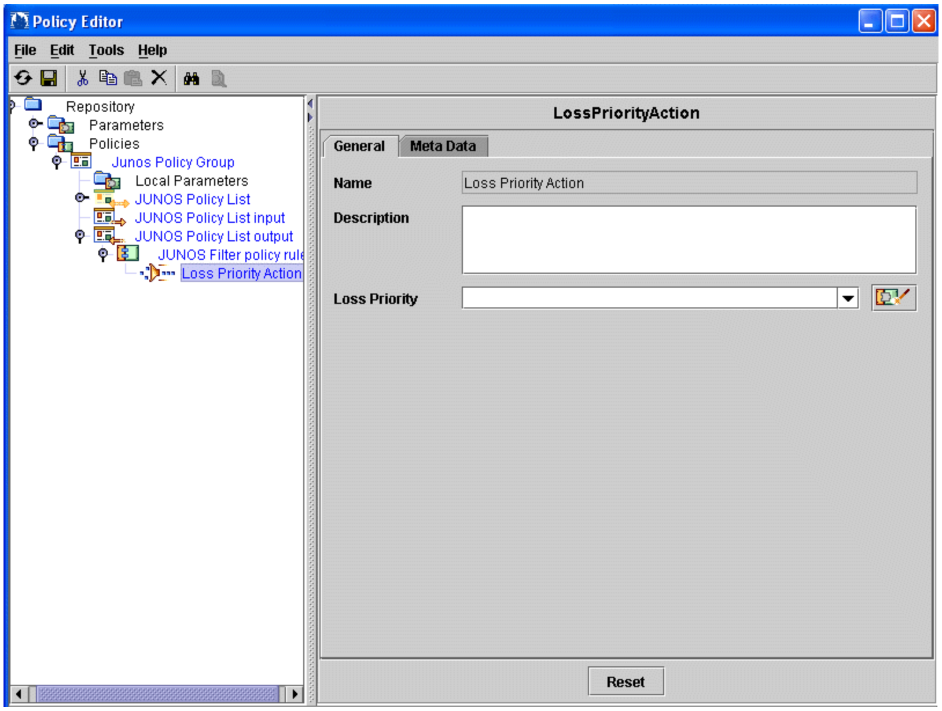
- Preemption bit in the session class ID. Use the preemption bit to allocate bandwidth to lower-priority sessions.
- Value
  - Number in the range 0–1
    - 0—Enables preemption
    - 1—Disables preemption
  - String expression
  - Parameter of type sessionClassIdPreemption
- Default—No value

Session Class Id Configurable

- Configurable bit in the session class ID. Application managers that provide novel services may use this value to specify new session classes. Use this field if your policy server supports configurable policies based on this value or if your CMTS device implements a novel session class based on this value.
- Value
  - Number in the range 0–15
  - String expression
  - Parameter of type sessionClassIdConfigurable
- Default—No value

Configuring Loss Priority Actions

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.



Name: Loss Priority Action

Description:

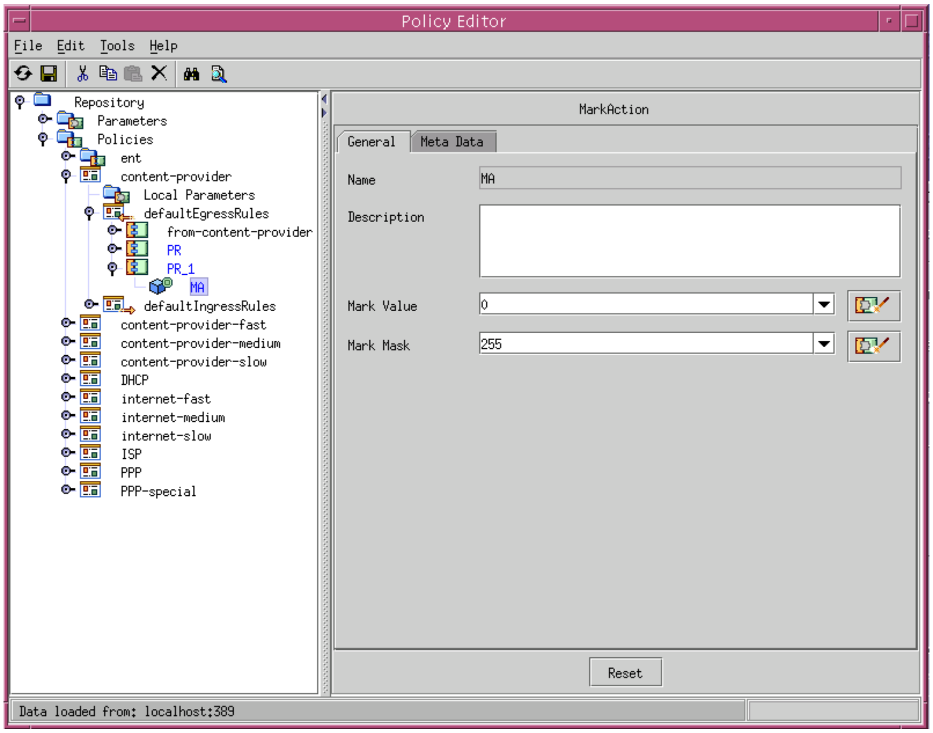
Loss Priority:

Loss Priority

- Sets the packet loss priority (PLP).
- Value
  - Predefined global parameter:
    - any_priority—Do not select this value for loss priority. This parameter appears in this field because it is a global packetLossPriority parameter. However, in this context, a value of any_priority is not valid.
    - high_priority—Sets the PLP to high
    - low_priority—Sets the PLP to low
  - String expression that matches valid values on the router; for example, “high” or “low”
  - Parameter of type packetLossPriority
- Default—No value

Configuring Mark Actions

Use this action to mark packets. You can configure mark actions for JUNOSe and PCMM policy rules.



Name: MA

Description:

Mark Value: 0

Mark Mask: 255

Data loaded from: localhost:389

Description

- Description of the action.
- Value—Text
- Default—No value

Mark Value

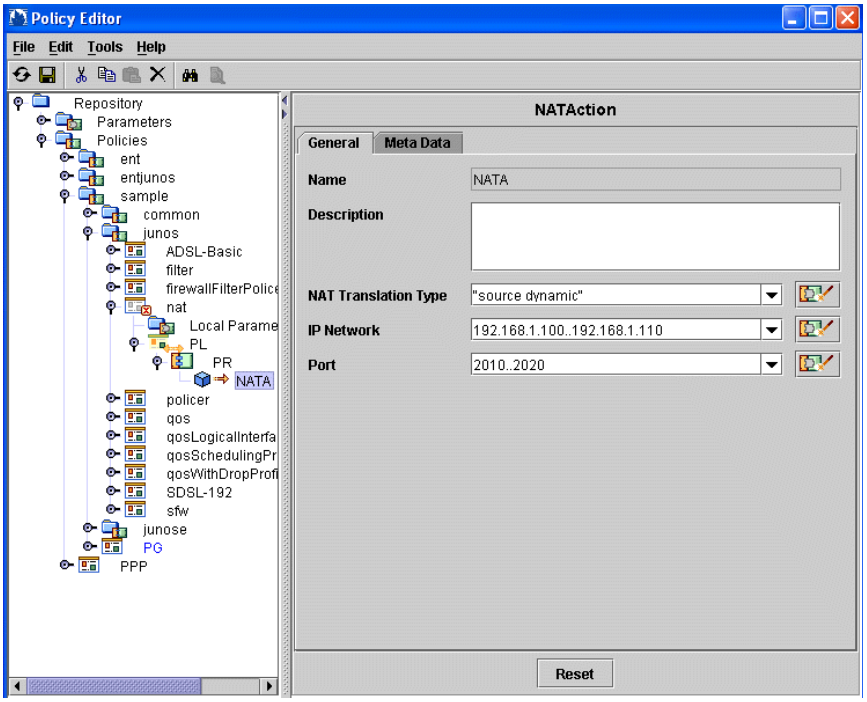
- For IPv4 packets, sets the ToS field in the IP header. For IPv6 packets, sets the traffic-class field in the IP header
- Value
  - Integer in the range 0–255
  - Parameter of type tosByte
- Default—0

Mark Mask

- Mask associated with the mark value.
- Value
  - Integer in the range 0–255
  - Parameter of type tosByteMask
- Default—255

Configuring NAT Actions

You can configure NAT actions for JUNOS ASP policy rules.



Name: NATA

Description:

NAT Translation Type: "source dynamic"

IP Network: 192.168.1.100..192.168.1.110

Port: 2010..2020

NAT Translation Type

- Type of network address translation that is used.
- Value
  - String expression that matches a NAT type on the router; for example:
    - “destination static”—Implements address translation for destination traffic without port translation; makes selected private servers accessible
    - “source dynamic”—Implements address translation for source traffic with port translation
    - “source static”—Implements address translation for source traffic without port mapping
  - Parameter of type natTranslationType
- Default—No value

IP Network

- IP address ranges.
- Value
  - An IP address with or without a prefix
  - Expression that indicates an address range (low to high); for example, 92.168.1.100..192.168.1.110; address ranges are limited to 32 addresses
  - Predefined global parameter:
    - any—Do not select this value for IP network. This parameter appears in this field because it is a global network parameter. However, in this context, a value of any is not valid.
  - Parameter of type network
  - Parameter of type address/prefix; for example, pubIp/32

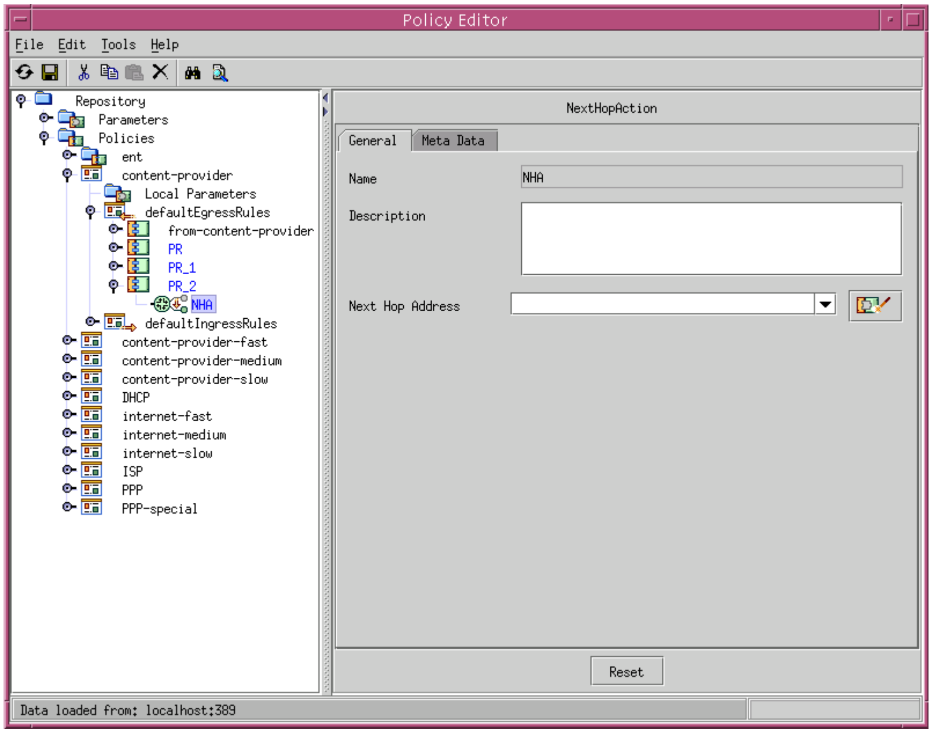
    where pubIp is a local address parameter and 32 is the prefix length
- Default—0.0.0.0/0.0.0.0

Port

- Port range to restrict port translation when NAT is configured in dynamic-source mode.
- Value
  - Predefined global parameter:
    - service_port—Port of the service as specified by the service object
  - Integer in the range 0–64000
  - Numeric expression that indicates a range of ports; for example, 2010..2020
  - 0..65535—Provides the same effect as the automatic option. JUNOS routing platforms support a port option called automatic, which means that it is a router-assigned port.
  - Parameter of type port
- Default—No value

Configuring Next-Hop Actions

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 JUNOSe policy rules.



Name: NHA

Description:

Next Hop Address:

Data loaded from: localhost:389

Description

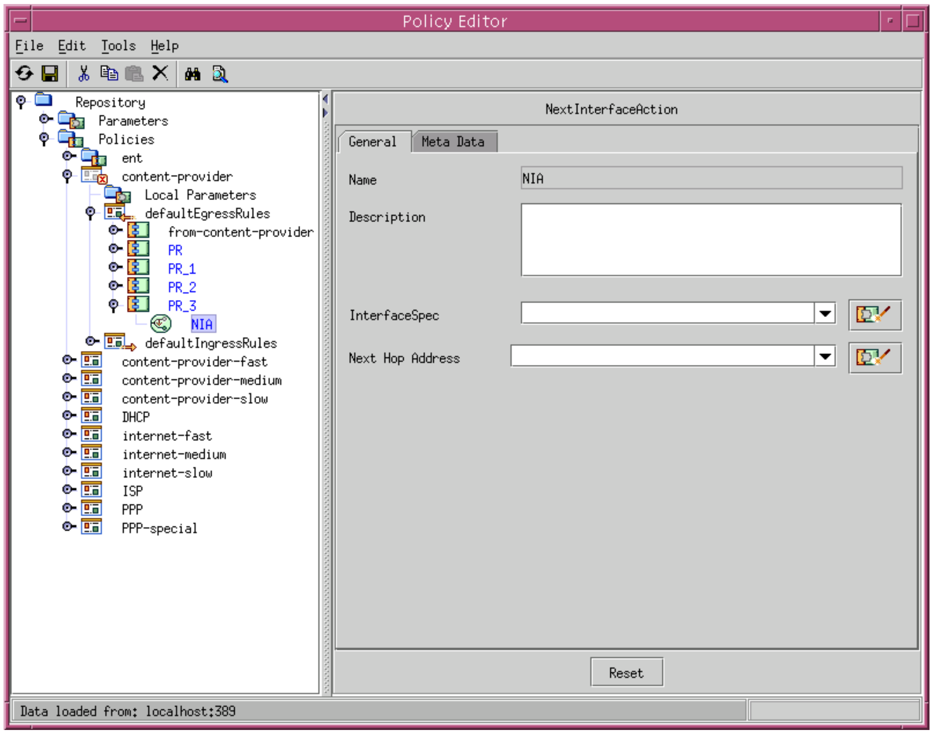
- Description of the action.
- Value—Text
- Default—No value

Next Hop Address

- Next IP address where the classified packets should go.
- Value
  - IP address
  - Predefined global parameter:
    - gateway_ipAddress—IP address of the gateway as specified by the service object
    - interface_ipAddress—IP address of the router interface
    - service_ipAddress—IP address of the service as specified by the service object
    - user_ipAddress—IP address of the subscriber
    - virtual_ipAddress—Virtual portal address of the SSP that is used in redundant redirect server installations
  - Parameter of type address
- Default—0

Configuring Next-Interface Actions

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 JUNOSe policy rules. On JUNOSe routers, you can use this action for both ingress and egress parts of the interface.



Name: NIA

Description:

Data loaded from: localhost:389

Description

- Description that is inherited from the managed element.
- Value—Text description
- Default—No value

InterfaceSpec

- IP interface to be used as the next interface for packets.
- Value
  - For JUNOSe interfaces:
    - Enter interface specifiers in the format:

      ‘<type of specifier> = <value>’

      where< type of specifier> is the interface name, alias, description, or uid

      For example: name='fastEthernet3/0'

      For lists of valid interface specifiers for JUNOSe routers, see *Interface Types and Specifiers* in the *JUNOSe Command Reference Guides*.
  - For JUNOS interfaces:
    - Enter interface specifiers in the format:

      ‘name= <mediatype>-<slot>/<pic>/<port>.<unit>’

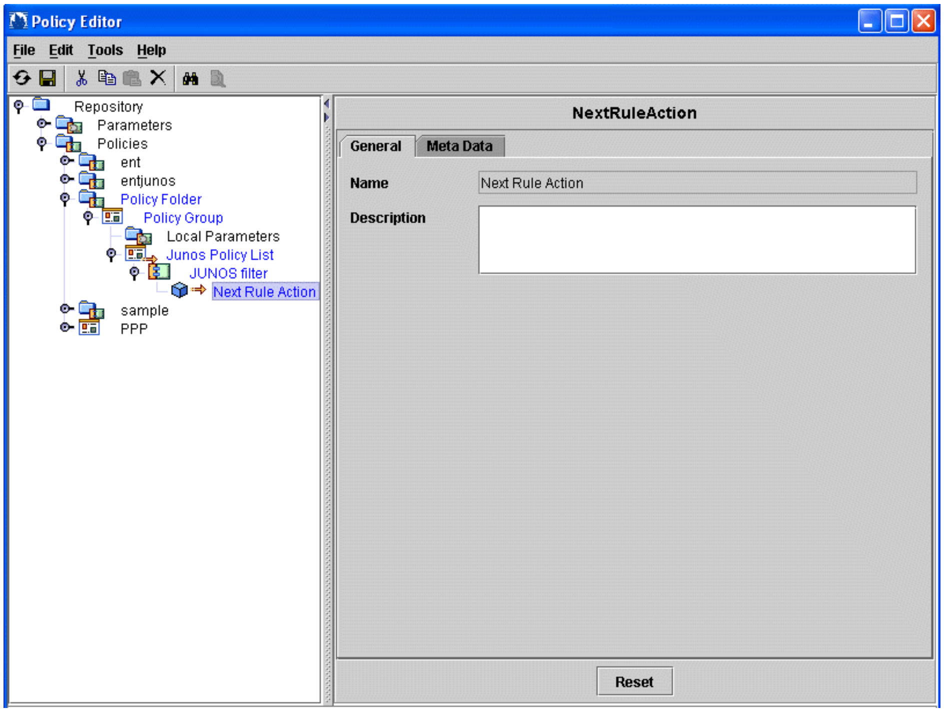
      For example: ‘name=AT-0/1/0.0’
  - Parameter of type interfaceSpec
- Default—No value

Next Hop Address

- Next IP address where the classified packets should go. This field is available only in JUNOSe policy rules.
- Value
  - IP address
  - Predefined global parameter:
    - gateway_ipAddress—IP address of the gateway as specified by the service object
    - interface_ipAddress—IP address of the router interface
    - service_ipAddress—IP address of the service as specified by the service object
    - user_ipAddress—IP address of the subscriber
    - virtual_ipAddress—Virtual portal address of the SSP that is used in redundant redirect server installations
  - Parameter of type address
- Default—No value

Configuring Next-Rule Actions

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.



Name: Next Rule Action

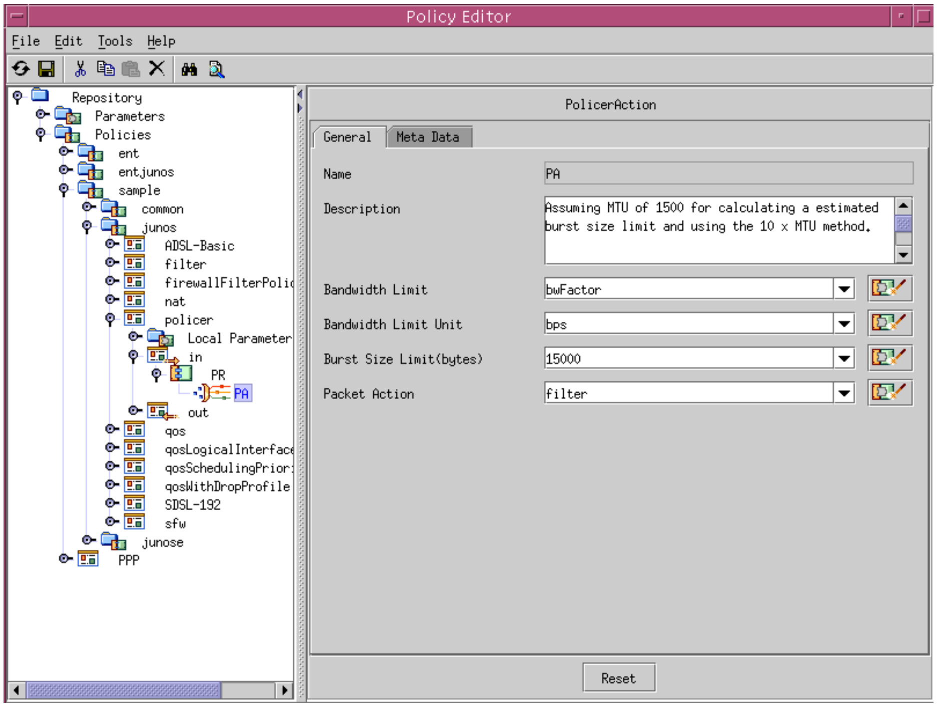
Description:

Description

- Description of the action.
- Value—Text
- Default—No value

Configuring Policer Actions

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.



Name: PA

Description: Assuming MTU of 1500 for calculating a estimated burst size limit and using the 10 x MTU method.

Bandwidth Limit: bwFactor

Bandwidth Limit Unit: bps

Burst Size Limit(bytes): 15000

Packet Action: filter

Description

- Description of the action.
- Value—Text
- Default—No value

Bandwidth Limit

- Traffic rate, that if exceeded, causes the router to take the indicated packet action.
- Value
  - Predefined global parameter:
    - interface_speed—Speed of the subscriber’s router interface
  - Integer that represents:
    - rate in bps
    - percentage of bandwidth
  - Numeric expression
  - Parameter of type rate

- Default—No value
- Example—bw * 1 / 5 sets a bandwidth limit of 1/5 of total bandwidth

  where bw is a local parameter that has a value of 1024 * 1920

Bandwidth Limit Unit

- Indicates the type of value entered for bandwidth limit.
- Value
  - Predefined global parameter:
    - bps—Value entered for bandwidth limit is bps
    - percent—Value entered for bandwidth limit is a percentage of the port speed
  - String expression
  - Parameter of type bandwidthSizeUnit
- Default—No value

Burst Size Limit (bytes)

- Maximum burst size. The minimum recommended value is the maximum transmission unit (MTU) of the IP packets being policed.
- Value
  - Number of bytes
  - Numeric expression; for example 8*64000
  - Parameter of type burst
- Default—No value
- Example—8*qosRate sets the burst size limit to 8 times the value of qosRate

  where qosRate is a local parameter of type rate

Packet Action

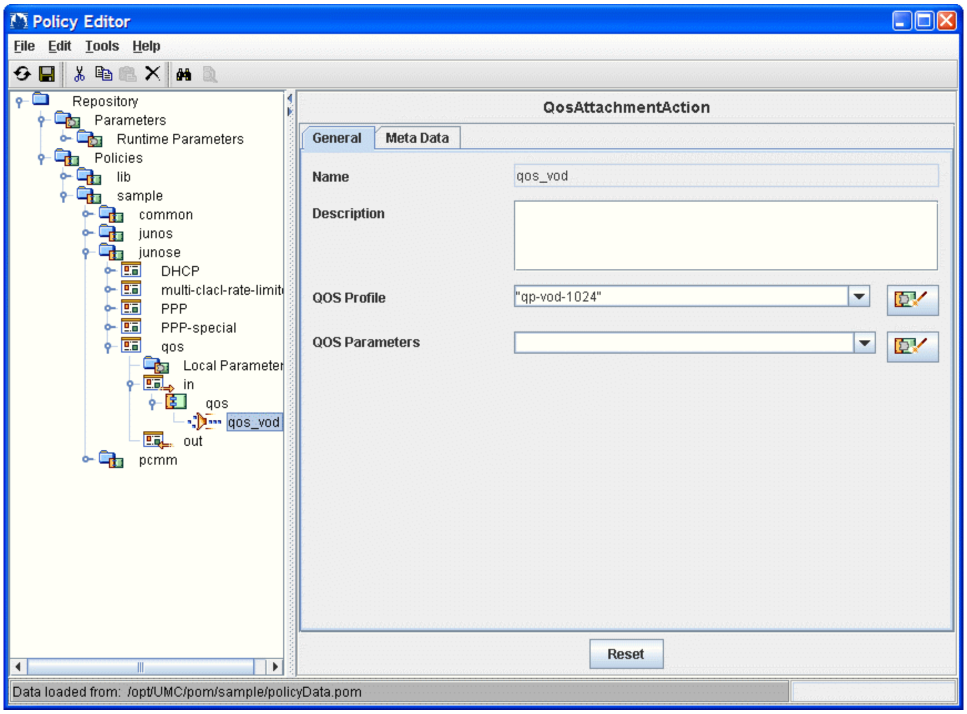
- Action taken on a packet that exceeds its rate limits.
- Value
  - filter—Packet is discarded
  - forwardingClass—Packet is assigned to a forwarding class
  - lossPriority—Packet's loss priority level is set to low or high
  - String expression
  - Parameter of type packetOperation
- Default—No value

Configuring QoS Profile Attachment Actions

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 JUNOSe 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 *SRC-PE Solutions Guide, Chapter 1, Managing Tiered and Premium Services with QoS on JUNOSe Routers with the SRC CLI*.

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.



QosAttachmentAction

Name: qos_vod

Description:

QOS Profile: "qp-vod-1024"

QOS Parameters:

Data loaded from: /opt/UMC/pom/sample/policyData.pom

Description

- Description of the action.
- Value—Text
- Default—No value

QoS Profile

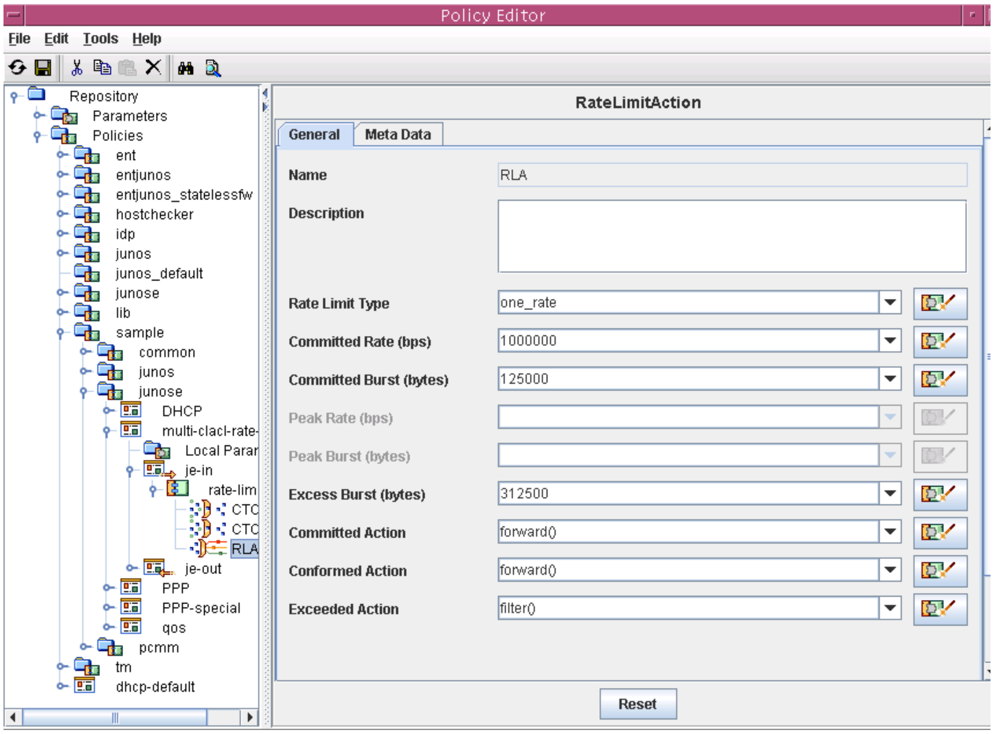
- Name of the QoS profile to attach to the JUNOSe interface when this action is taken.
- Value
    - Name of a QoS profile that is configured on the router. Enclose the name in double quotation marks to indicate that it is a literal string and not a parameter.
    - Parameter of type qosProfileSpec
- Default—No value

QoS Parameters

- Names and values of the QoS parameters to attach to the JUNOSe interface when this action is taken. The parameters are configured on the JUNOSe router and referenced in the scheduler profiles referred to by the QoS profile.
- Value
    - The name = value pair that defines a QoS parameter; use map expressions to define multiple QoS parameters. For example, the map expression {max-bw = 512000, shape-rate = 1000000} supplies two QoS parameters.
    - Parameter of type map
- Default—No value

Configuring Rate-Limit Actions

Use this action to define the quality of service. You can configure rate-limit actions for JUNOSe policy rules.



Name: RLA

Description:

Rate Limit Type: one_rate

Committed Rate (bps): 1000000

Committed Burst (bytes): 125000

Peak Rate (bps):

Peak Burst (bytes):

Excess Burst (bytes): 312500

Committed Action: forward()

Conformed Action: forward()

Exceeded Action: filter()

Description

- Description of the profile.
- Value—Text
- Default—No value

Rate Limit Type

- Specifies that the rate-limit profile is either one rate or two rate. The one-rate rate-limit profile provides a hard-limit rate limiter or a TCP-friendly rate limiter. The two-rate rate-limit profile provides a two-rate, three-color marking mechanism.

- Value
  - Predefined global parameter:
    - one rate—Uses a single-rate committed rate with two burst parameters: committed burst and excess burst; supports a TCP-friendly rate limiter
    - two rate—Uses committed rate and peak rate, each with a burst parameter
  - Parameter of type rateLimitType
- Default—Two rate

Committed Rate (bps)

- Target rate for the traffic that the policy covers.
- Value
  - Predefined global parameter:
    - interface_speed—Speed of the subscriber's router interface
  - Number of bits per second in the range 0–4294967295
  - Parameter of type rate
- Default—0

Committed Burst (bytes)

- Amount of bandwidth allocated to burst traffic in bytes.
- Value
  - Number of bytes in the range 8192–4294967295
  - Numeric expression
  - Parameter of type burst
- Default—16384
- Example—max(qos*0.1/8, 16384) – sets the burst size to the maximum of 100-ms burst at committed rate (qos*0.1) in bytes (/8) or 16384

  where qos is a local parameter that represents the committed rate

Peak Rate (bps)

- For two-rate rate-limit profiles, specifies the amount of bandwidth allocated to excess traffic flow over the committed rate.
- Value
  - Predefined global parameter:
    - interface_speed—Speed of the subscriber's router interface
  - Number of bits per second in the range 0–4294967295
  - Numeric expression
  - Parameter of type rate
- Default—0
- Example—qos*1.5 – sets the peak rate to 1.5 times the committed rate

  where qos is a local parameter that represents the committed rate

Peak Burst (bytes)

- For two-rate rate-limit profiles, specifies the amount of bandwidth allocated to burst traffic in excess of the peak rate.
- Value
  - Number of bytes in the range 8192–4294967295
  - Numeric expression
  - Parameter of type burst
- Default—16384
- Example—max(qos*1.5*0.1/8, 16384)

  where qos is a local parameter that represents the committed rate

Excess Burst (bytes)

- For one-rate rate-limit profiles, specifies the amount of bandwidth allocated to accommodate burst traffic.
- Value
  - Number of bytes in the range <0 |[Committed Burst + 1, 4294967295]>
  - Numeric expression
  - Parameter of type burst
- Default—No value

Committed Action

- Policy action if traffic flow does not exceed the committed rate.
- Value
  - filter()—Drops the packet
  - forward()—Transmits the packet
  - mark()—Marks the packet by setting the ToS byte (IP) or traffic-class field (IPv6) to the specified 8-bit value, and transmits the packet. Specify the ToS byte in the parenthesis.

    The ToS byte can be an integer in the range 0–255 or parameter of type tosByte
  - Parameter of type packetOperation
- Default—Forward

Conformed Action

- Policy action if traffic flow exceeds the committed rate but remains below the peak rate.
- Value
  - filter()—Drops the packet
  - forward()—Forwards the packet
  - mark()—Marks the packet by setting the ToS byte (IP) or traffic-class field (IPv6) to the specified 8-bit value, and transmits the packet. Specify the ToS byte in the parenthesis.

    The ToS byte can be an integer in the range 0–255 or parameter of type tosByte
  - Parameter of type packetOperation
- Default—Forward

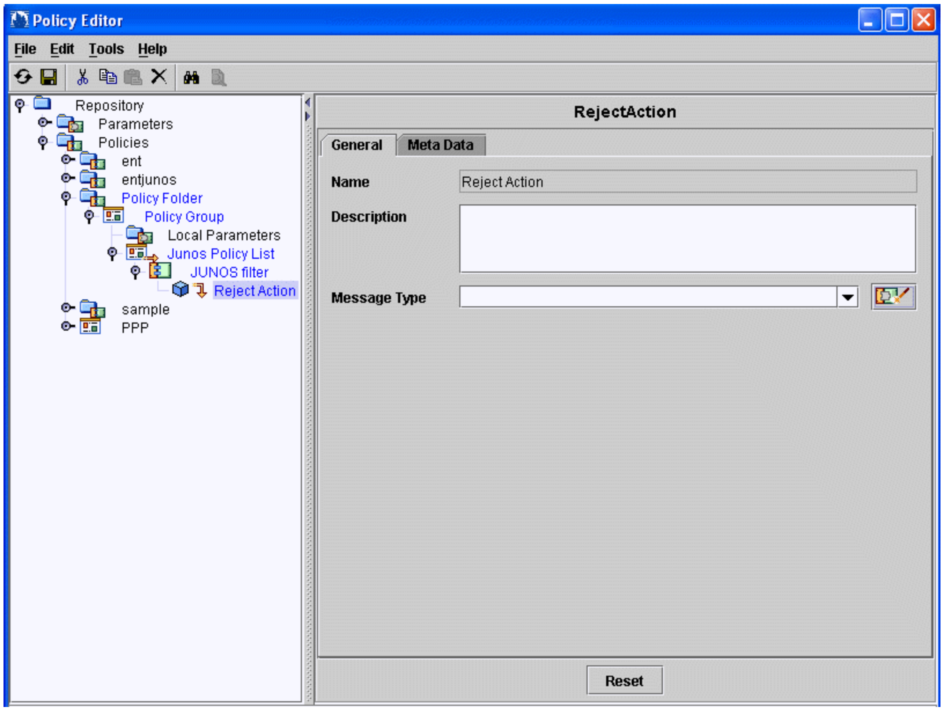
Exceeded Action

- Policy action if traffic flow exceeds the peak rate.
- Value
  - filter()—Drops the packet
  - forward()—Transmits the packet
  - mark()—Marks the packet by setting the ToS byte (IP) or traffic-class field (IPv6) to the specified 8-bit value, and transmits the packet. Specify the ToS byte in the parenthesis.

    The ToS byte can be an integer in the range 0–255 or parameter of type tosByte
  - Parameter of type packetOperation
- Default—Forward

Configuring Reject Actions

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.



Name: Reject Action

Description

- Description of the action.
- Value—Text
- Default—No value

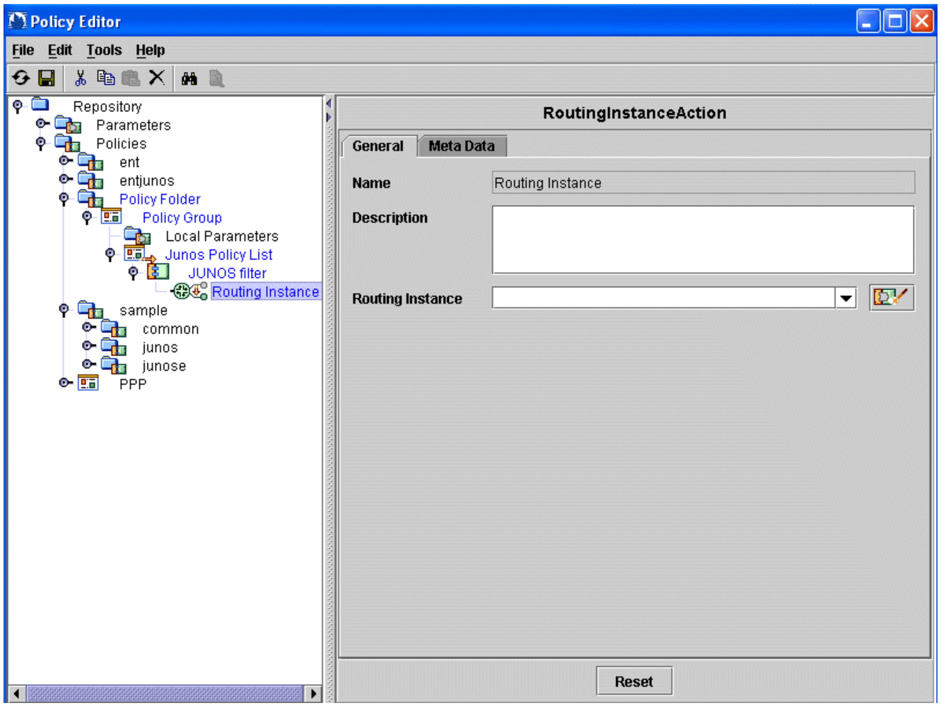
Message Type

- Type of ICMP destination unreachable message sent to the client.
- Value
  - String expression that matches a type of ICMP destination unreachable message supported on the router; for example:
    - “administratively-prohibited”
    - “bad-host-tos”
    - “bad-network-tos”
    - “host-prohibited”
    - “host-unknown”
    - “host-unreachable”
    - “network-prohibited”

        - "network-unknown"
        - "network-unreachable"
        - "port-unreachable"
        - "precedence-cutoff"
        - "precedence-violation"
        - "protocol-unreachable"
        - "source-host-isolated"
        - "source-route-failed"
        - "tcp-reset"—If you specify tcp-reset, a TCP reset message is sent if the packet is a TCP packet. Otherwise, nothing is sent.
    - Parameter of type messageType
- Default—No value

Configuring Routing Instance Actions

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.



RoutingInstanceAction

Name: Routing Instance

Description:

Routing Instance:

Description

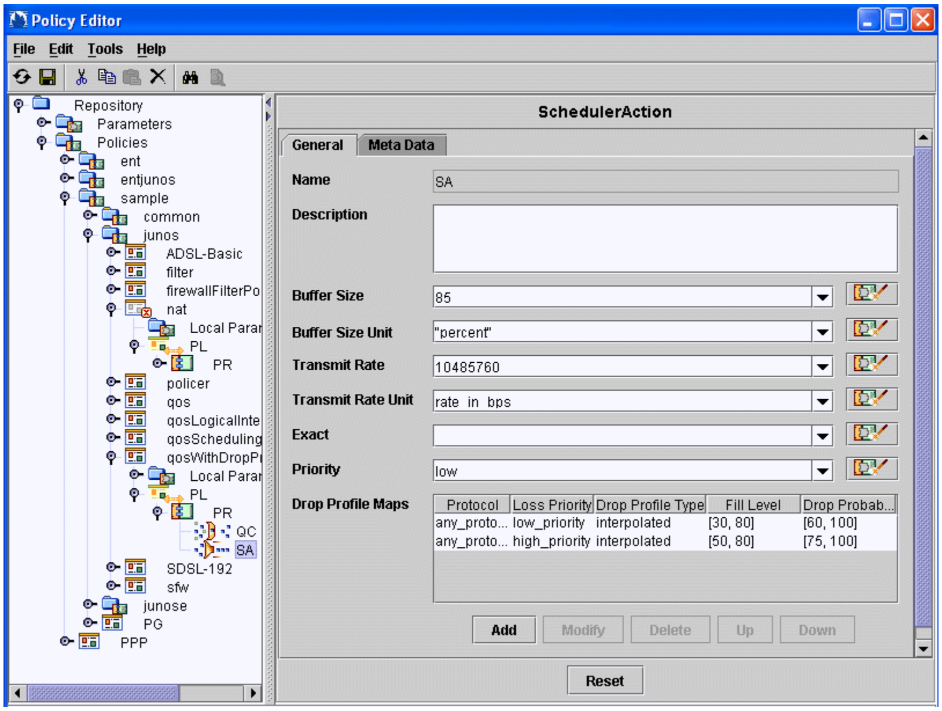
- Description of the action.
- Value—Text
- Default—No value

Routing Instance

- Routing instance to which packets are forwarded. The routing instance must be configured on the router.
- Value
  - String expression that matches the name of a routing instance configured on the router; for example “isp2-route-table”
  - Parameter of type routingInstance
- Default—No value

Configuring Scheduler Actions

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.



Name: SA

Description:

Buffer Size: 85

Buffer Size Unit: "percent"

Transmit Rate: 10485760

Transmit Rate Unit: rate in bps

Exact:

Priority: low

Drop Profile Maps

Protocol	Loss Priority	Drop Profile Type	Fill Level	Drop Probab...
any_proto...	low_priority	interpolated	[30, 80]	[60, 100]
any_proto...	high_priority	interpolated	[50, 80]	[75, 100]

Description

- Description of the action.
- Value—Text
- Default—No value

Buffer Size

- Buffer size.
- Value
  - Integer that represents:
    - microseconds
    - percentage of total buffer size
  - “remainder”—Uses available buffer that is not assigned to other queues
  - Expression
  - Parameter of type schedulerBufferSize
- Default—No value

Buffer Size Unit

- Indicates the type of value that you entered for buffer size.
- Value
  - Predefined global parameter:
    - buffer_size_percentage—The value is a percentage of the total buffer.
    - buffer_size_remainder—The value is the remaining buffer available.
    - temporal—The value is temporal, in microseconds.
  - String expression; for example, “percent”
  - Parameter of type schedulerBufferSizeUnit
- Default—No value

Transmit Rate

- Transmit rate.
- Value
  - Integer that represents:
    - Rate in bps
    - Percentage of bandwidth
  - “remainder”—Uses remaining rate available
  - Numeric expression
  - Parameter of type schedulerTransmitRate
- Default—No value
- Example—4/10*bandwidth sets the transmit rate to 4/10 of transmission bandwidth that is allocated to the logical interface unit

  where bandwidth is a local parameter of type any

Transmit Rate Unit

- Indicates the type of value entered for transmit rate.
- Value
  - Predefined global parameter:
    - rate_in_bps—Transmission rate in bps
    - rate_in_percentage—Percentage of transmission capacity
    - rate_in_remainder—Uses remaining rate available
  - String expression
  - Parameter of type schedulerTransmitRateUnit
- Default—No value

Exact

- Specifies 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.
- Value
  - true
  - false
  - Parameter of type boolean
- Default—No value

Priority

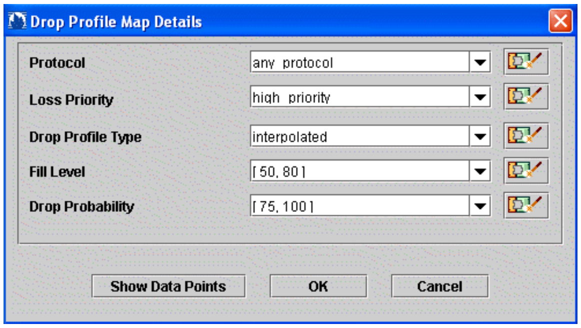
- Packet-scheduling priority. The priority determines the order in which an output interface transmits traffic from the queues.
- Value
  - Predefined global parameter:
    - low
    - medium_low
    - medium_high
    - high—Assigning high priority to a queue prevents the queue from being starved by traffic in a strict high-priority queue
    - strict_high—Configure a high-priority queue with unlimited transmission bandwidth available to it. As long as it has traffic to send, the strict high-priority queue receives precedence over low, medium-low, and medium-high priority queues, but not high-priority queues. You can configure strict high-priority on only one queue per interface.
  - String expression—For example, “strict-high”
  - Parameter of type schedulerPriority
- Default—No value

Configuring Drop Profile Maps

The scheduler drop profile defines the drop probabilities across the range of delay-buffer occupancy, thereby supporting the RED process. 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.

The SchedulerAction pane displays a table with configured drop profile maps. To configure the table:

- To add a drop profile map, click **Add**. Policy Editor displays the Drop Profile Map Details dialog box.
- To modify a map, select the map, and click **Modify**. Policy Editor displays the Drop Profile Map Details dialog box for that map.
- To delete a map, select the map, and click **Delete**.
- To move a map up, select the map, and click **Up**.
- To move a map down, select the map, and click **Down**.



Drop Profile Map Details

Protocol: any protocol
Loss Priority: high priority
Drop Profile Type: interpolated
Fill Level: [50, 80]
Drop Probability: [75, 100]

Protocol

- Specify the protocol type for the drop protocol.
- Value
  - Predefined global parameter:
    - any_protocol—Accepts any protocol type
    - non_tcp—Accepts any protocol type other than TCP/IP
    - tcp_only—Accepts only TCP/IP protocol
  - String expression
  - Parameter of type dropProfileProtocol
- Default—No value

Loss Priority

- Sets the packet loss priority (PLP).
- Value
  - Predefined global parameter:
    - any_priority—Drop profile applies to packets with any PLP.
    - high_priority—Drop profile applies to packets with high PLP.
    - low_priority—Drop profile applies to packets with low PLP.
  - String expression
  - Parameter of type packetLossPriority
- Default—No value

Drop Profile Type

- Relationship between the fill level and drop probability.
- Value
  - Predefined global parameter:
    - interpolated—Specifies values for interpolating relationship between queue fill level and drop probability
    - segmented—Specifies fill level and drop probability as percentages
  - Parameter of type dropProfileType
- Default—No value

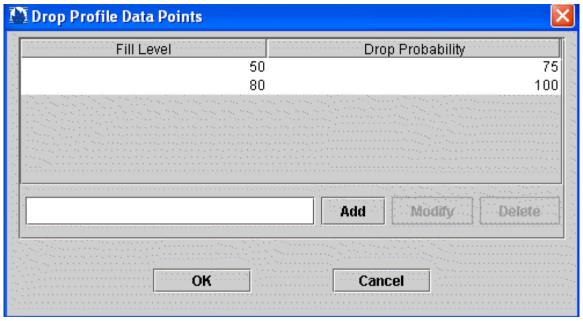
Setting Fill Level and Drop Probability

In drop profiles you configure fill level 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.

You can set these value pairs by clicking Show Data Points on the Drop Profile Map Details screen. To add a value pair:

1. In the data entry field, enter the value for the fill level, press the space bar, and then enter the drop probability value.

2. Click **Add**.



Fill Level	Drop Probability
50	75
80	100

Fill Level

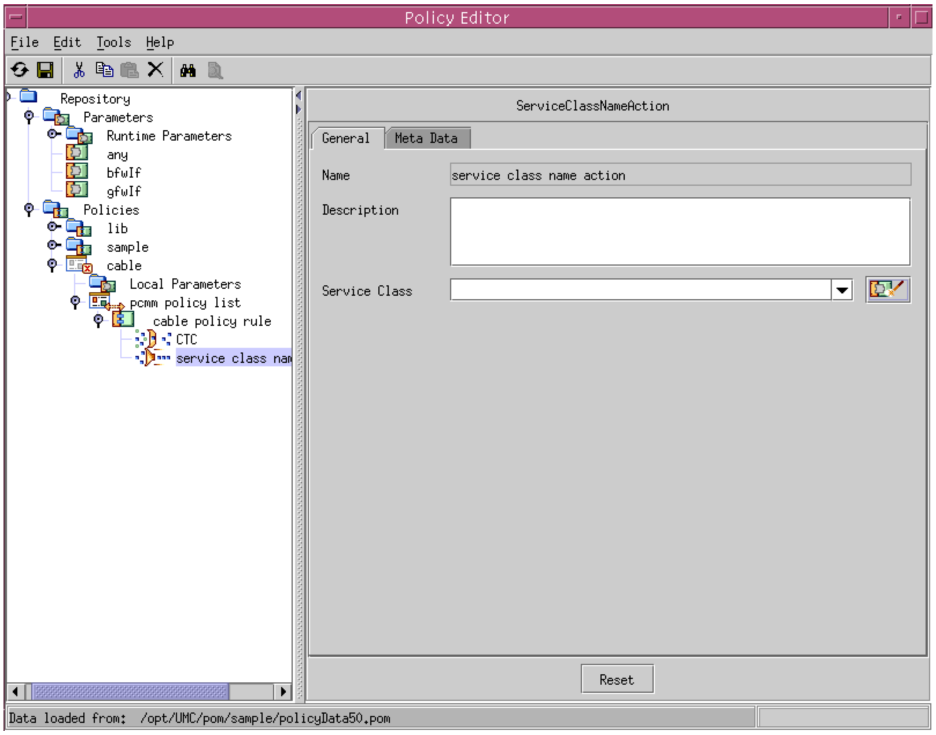
- Fill level of the queue.
- Value
  - If the drop profile type is segmented, specify how full the queue is as a percentage.
  - If the drop profile type is interpolated, specify a data point for mapping the queue fill percentage in the range 0–100.
  - Parameter of type percent
- Default—No value

Drop Probability

- Probability that a packet will be dropped.
- Value
  - If the drop profile type is segmented, specify the drop probability as a percentage. A value of 0 means that a packet will never be dropped, and a value of 100 means that all packets will be dropped. The range is 0–100.
  - If the drop profile type is interpolated, specify a data point for packet drop probability in the range 0–100.
  - Parameter of type percent
- Default—No value

Configuring Service Class Name Actions

You can configure service class name actions for PCMM policy rules.



Name: service class name action

Description:

Service Class:

Data loaded from: /opt/UMC/pom/sample/policyData50.pom

Description

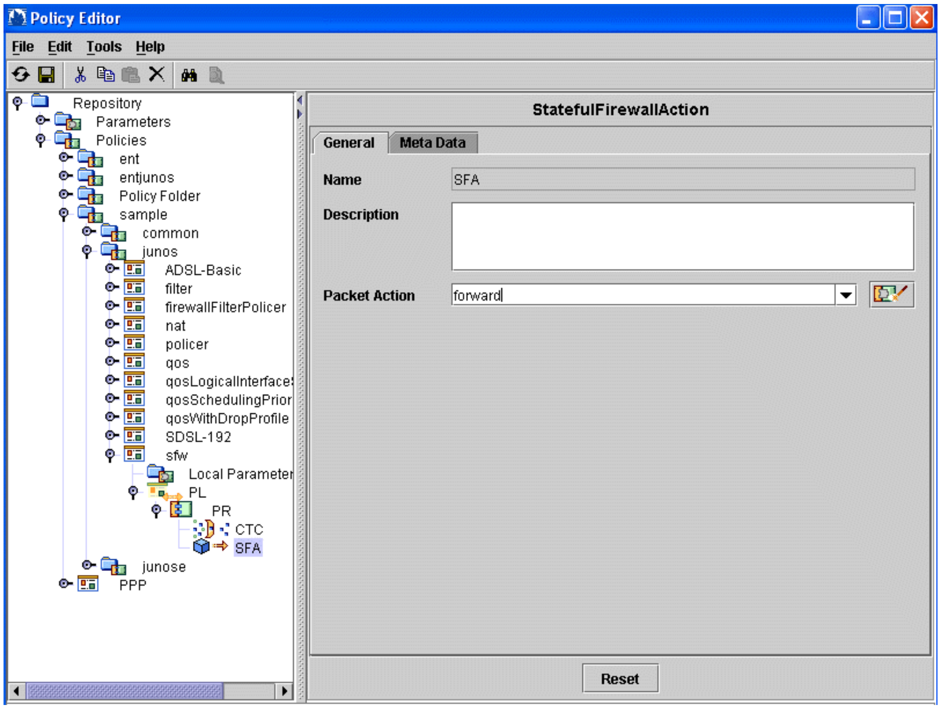
- Description of the action.
- Value—Text
- Default—No value

Service Class

- Name of a service class on the CMTS device that specifies QoS parameters for a service flow.
- Value
  - Name of a service class
  - String expression
  - Parameter of type serviceClassName
- Default—No value

Configuring Stateful Firewall Actions

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.



Name: SFA

Description:

Packet Action: forward

Description

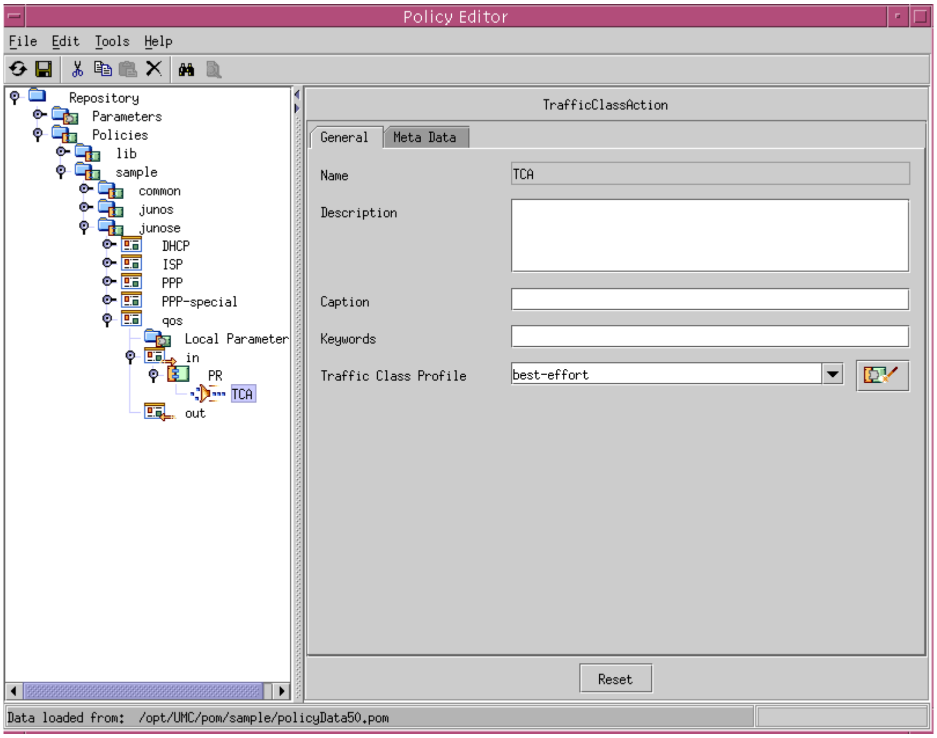
- Description of the action.
- Value—Text
- Default—No value

Packet Action

- Action taken on packets.
- Value
  - filter—Packet is not accepted and is not processed further
  - forward—Packet is accepted and sent to its destination
  - reject—Packet is not accepted, and a rejection message is returned; UDP sends an ICMP unreachable code, and TCP sends RST
  - String expression
  - Parameter of type packetOperation
- Default—No value

Configuring Traffic-Class Actions

Use this action to put packets in a particular traffic class. You can configure traffic-class actions for JUNOSe policy rules.



Name: TCA

Description:

Caption:

Keywords:

Traffic Class Profile: best-effort

Data loaded from: /opt/UMC/pom/sample/policyData50.pom

Description

- Description of the action.
- Value—Text
- Default—No value

Caption

- Short description of the action.
- Value—Text
- Default—No value

Keywords

- Series of words that the system uses as a filter for keyword searches that are inherited from the policy.
- Value—Text
- Default—No value

Traffic-Class Profile

- Name of the traffic-class profile that is applied to a packet when it passes through the router.
- Value
  - Name of a traffic-class profile that is configured on the router
  - Parameter of type trafficClassSpec
- Default—No value

Configuring Traffic-Mirror Actions

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 only for JUNOS input policy rules.

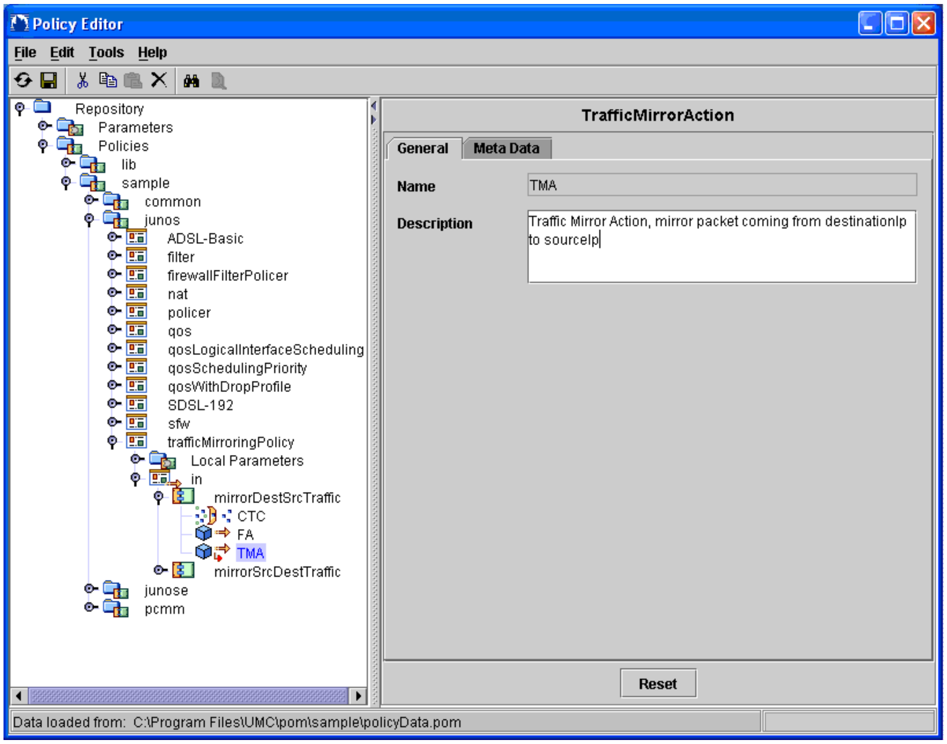
Before you use traffic-mirror actions, you must configure forwarding options on JUNOS routing platforms for port mirroring and next-hop group. For information about how these features work on the router, see the *JUNOS Policy Framework Configuration Guide*.

The 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.



Name: TMA

Description: Traffic Mirror Action, mirror packet coming from destinationIp to sourceIp

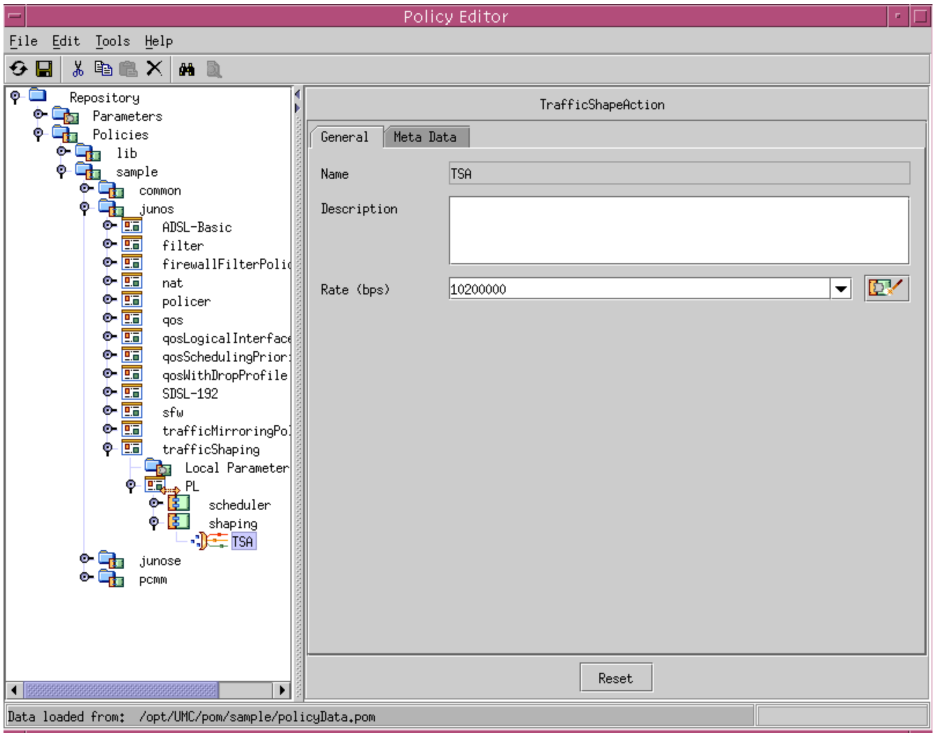
Data loaded from: C:\Program Files\UMC\pom\sample\policyData.pom

Description

- Description of the action.
- Value—Text
- Default—No value

Configuring Traffic-Shape Actions

Traffic-shape actions specify the maximum rate of traffic transmitted on an interface. You can create traffic-shape actions in JUNOS shaping policy rules.



Name: TSA

Description:

Rate (bps): 10200000

Data loaded from: /opt/UMC/pom/sample/policyData.pom

Description

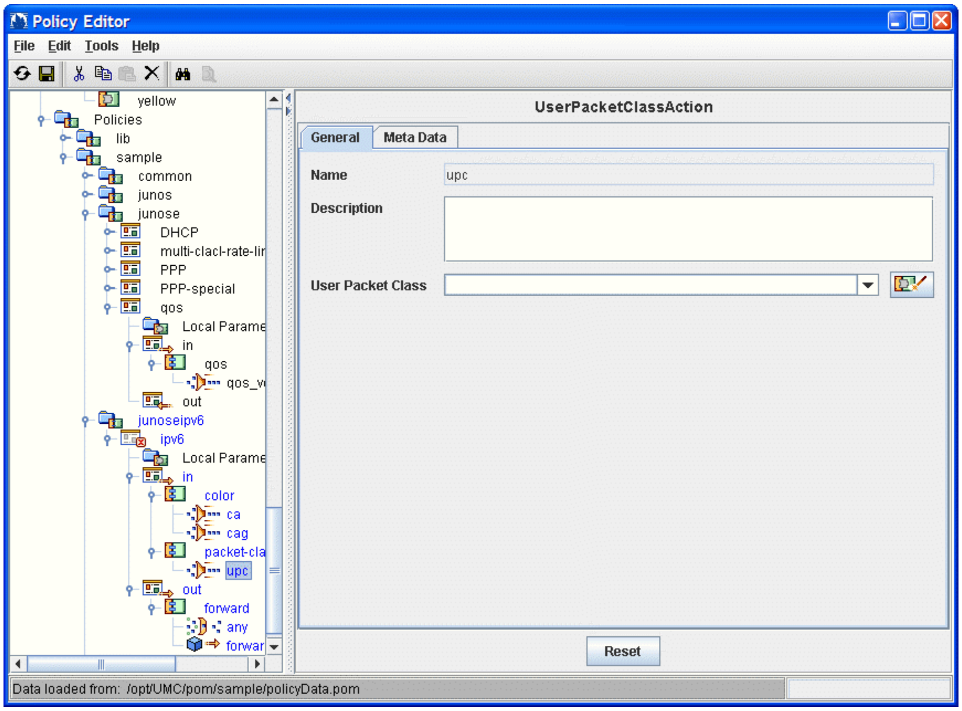
- Description of the action.
- Value—Text
- Default—No value

Rate (bps)

- Sets the maximum transmission rate.
- Value
  - Predefined global parameter:
    - interface_speed—Speed of the subscriber's router interface
  - Bits per second in the range 1000–32000000000
  - Numeric expression
  - Parameter of type rate
- Default—No value

Configuring User Packet Class Actions

Use this action to put packets in a particular user packet class. You can configure user packet class actions for JUNOSe IPv6 policy rules.



Name: upc

Description:

User Packet Class:

Data loaded from: /opt/UMC/pom/sample/policyData.pom

Description

- Description of the action.
- Value—Text
- Default—No value

User Packet Class

- User packet class that is applied to a packet when it passes through the router.
- Value
  - User packet class value that is configured on the router, in the range 0–15
  - Parameter of type userPacketClass
- Default—No value

Modifying Policy Objects in the Directory

This section shows how to modify policy groups by changing the policyGroup, policyList, and policyRule objects in the directory.

Once a policy is in use, we recommend that you do not modify the policy by deleting and recreating it. Doing so results in an error message being logged for each interface or active service session that currently uses the policy. If you delete a default policy that is running on an interface, the SRC software leaves the policy running and logs an error message. When a new interface that uses the policy as a default policy is created, every service activation for a service that uses the policy fails until the new definition of the policy is loaded. This condition lasts until DES polls the directory, detects the change, and provides the change to the policy engine.

Modifying Policy Groups

To modify an existing policyGroup and trigger the policy engine to update policies on JUNOSe routers:

1. Make the required changes to the policyList and policyRule objects that are contained in the policyGroup entry.

2. Make a modification to the policyGroup entry. For instance, change its description or set its deleted attribute to FALSE.

   This step triggers the policy engine to reload the new policy definition. All interfaces that currently use the policy as a default policy are updated, and all active service sessions that use the policy are updated.

Adding Policy Groups

To add a policy group and load it onto the JUNOSe router:

1. Make sure that a policyGroup object with the same name does not already already exist with its deleted attribute set to TRUE.

2. Create the policyGroup, and set the deleted attribute to TRUE.

3. Configure the policyGroup as desired, and configure its policyLists and policyRules.

4. Trigger the policy engine to load the new policy by setting the deleted attribute in the policyGroup to FALSE.

Deleting and Purging Policy Groups from the Directory

To delete a policyGroup entry from the directory, make sure that the umcDeletionAuxClass is in the object class, and set the deleted attribute to TRUE. At the next DES polling interval, the policy is removed from the policy engine. As mentioned above, take care not to delete policyGroups that are in use.

After you set the deleted attribute in the policyGroup to TRUE, you can purge the policyLists and policyRules underneath the policyGroup. Once you are sure that the deletion of policyLists and policyRules is replicated to all directories and that the SAE has been triggered to make the change, you can purge the policyGroup.

We recommend that you purge only deleted policyGroups. You can perform this operation very infrequently (perhaps once a month). Before performing this operation, use SAE Web Admin to check each SAE to be sure that the policyGroups to be purged are not included in the SAE's memory. If a deleted policy remains in the SAE's memory, ensure that it has its deleted attribute set to TRUE or that it does not exist in the SAE's connected directory. If the deleted policy:

- Has its deleted attribute set to TRUE, use SAE Web Admin to reload the policies from the directory.

- Does not exist in the SAE's connected directory, directory replication is not working properly and should be checked.

Chapter 12

Policy Examples Created with the SRC CLI

This chapter gives examples of policies that service providers can use to provide Internet access and to deploy different types of services. The examples in this chapter are created with the SRC CLI. To see these examples created with Policy Editor, see *Chapter 13, Policy Examples Created with Policy Editor*.

Topics in this chapter include:

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.

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;
        }
      }
    }
  }
```

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 1Mbps.
- 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.

Types of Policies

JUNOSe 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 JUNOSe Rate-Limiting Policy

The sample JUNOSe 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 JUNOSe 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 1Mbps; 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";
}
```

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.

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 *Sample Access Policies* on page 368). 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;
}
```

Chapter 13

Policy Examples Created with Policy Editor

This chapter gives examples of policies that service providers can use to provide Internet access and to deploy different types of services. The examples in this chapter are created with Policy Editor. To see these examples created with the SRC CLI, see *Chapter 12, Policy Examples Created with the SRC CLI*.

Topics in this chapter include:

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.

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 SSP 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

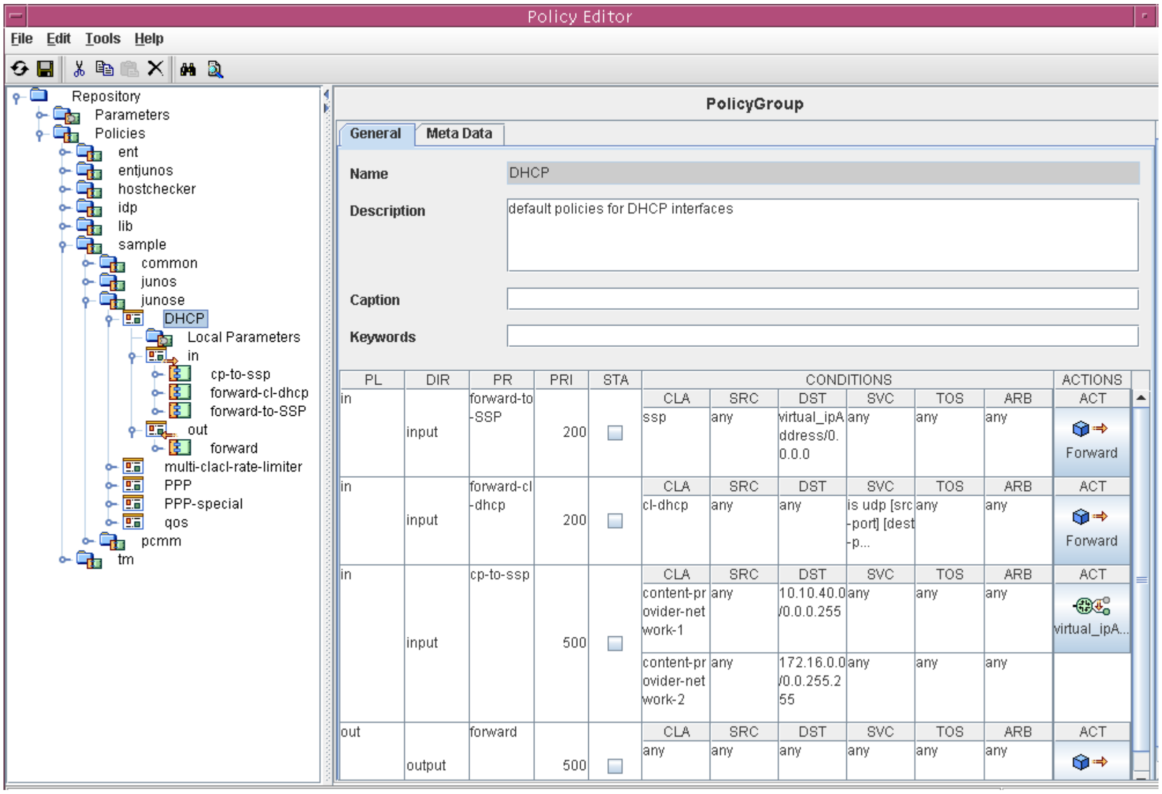
This section contains examples of access policies for DHCP subscribers and PPP subscribers. In both of these examples, there are two content providers. Traffic destined for the content provider networks is sent to the residential portal by using a next-hop action that forwards traffic to the virtual IP address of the SSP. (See *SRC-PE Sample Applications Guide*.)

Traffic to the SSP has a high priority and is not affected by other service policies. This way, the subscriber can always access the SSP. Traffic from the network is forwarded without any restrictions.

DHCP Policy Group

Figure 32 shows a summary of the access policy for DHCP subscribers.

Figure 32: DHCP Policy Group



Name: DHCP

Description: default policies for DHCP interfaces

Caption:

Keywords:

PL	DIR	PR	PRI	STA	CONDITIONS						ACTIONS
					CLA	SRC	DST	SVC	TOS	ARB	ACT
in	input	forward-to-SSP	200	☐	ssp	any	virtual_ipAddress/0.0.0.0	any	any	any	Forward
in	input	forward-cl-dhcp	200	☐	cl-dhcp	any	any	is udp [src-port] [dest-p...	any	any	Forward
in	input	cp-to-ssp	500	☐	content-provider-network-1	any	10.10.40.0/0.0.0.255	any	any	any	virtual_ipA...
					content-provider-network-2	any	172.16.0.0/0.0.255.255	any	any	any	
out	output	forward	500	☐	any	any	any	any	any	any	

The following information shows the configuration details of the DHCP policy group in Figure 32.

policyGroupName=DHCP, ou=junose, ou=sample, o=Policies, o=umc

PolicyList out

```
        name=out
        policyRoles=JUNOSE
        applicability=output

   PolicyRule forward
        priority=500
        type=JUNOSE
        accountingRule=false
     ClassifyTrafficCondition
     Forward Action
```

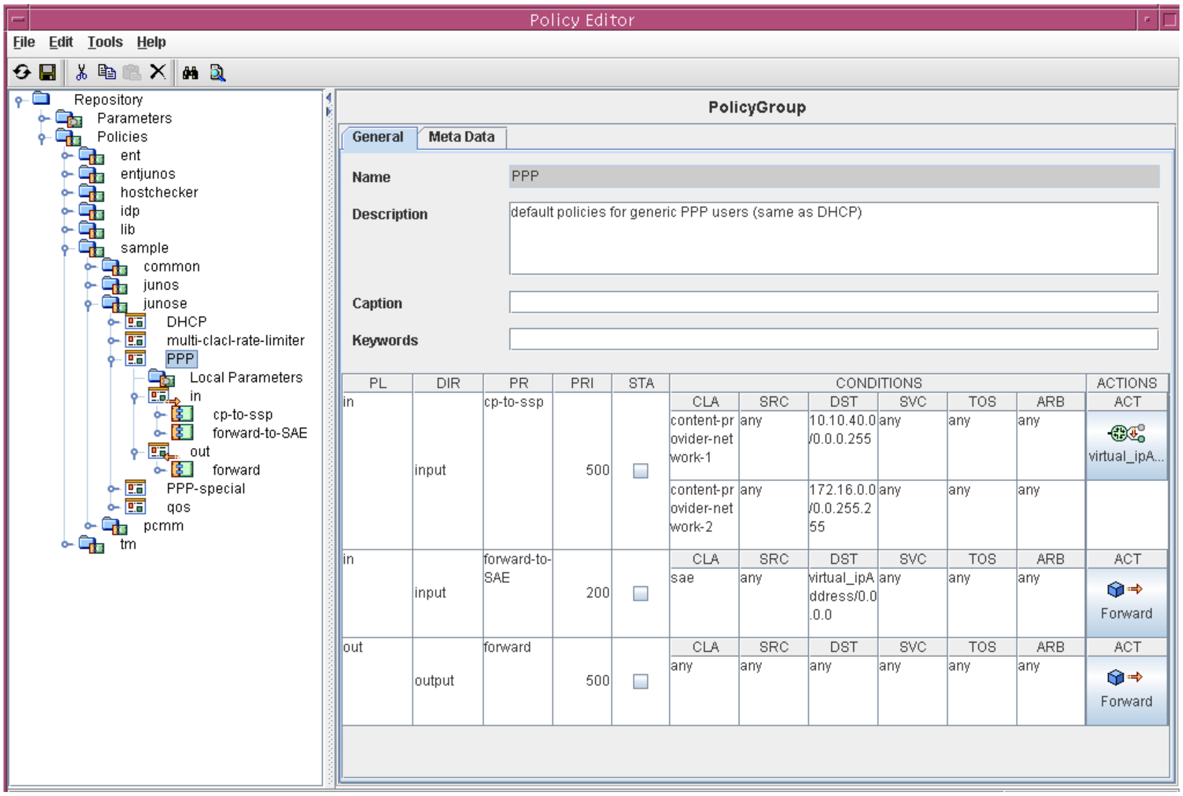
PolicyList in

```
        name=in
        policyRoles=JUNOSE
        applicability=input

   PolicyRule cp-to-ssp
        priority=500
        type=JUNOSE
        accountingRule=false
     ClassifyTrafficCondition content-provider-network-1
        DestinationNetwork:
          ipAddress=10.20.40.0
          ipMask=255.255.255.0
          ipOperation=is
     ClassifyTrafficCondition content-provider-network-2
        DestinationNetwork:
          ipAddress=172.16.0.0
          ipMask=0.0.255.255
          ipOperation=is
     NextHop Action
        nextHopAddress=virtual_ipAddress

   PolicyRule forward-cl-dhcp
        priority=200
        type=JUNOSE
        accountingRule=false
     ClassifyTrafficCondition
        protocol=udp
        DestinationNetwork:
          ipAddress=0.0.0.0
          destination port=67
     Forward Action

   PolicyRule forward-to-ssp
        priority=200
        type=JUNOSE
        accountingRule=false
     ClassifyTrafficCondition
        DestinationNetwork:
          ipAddress=virtual_ipAddress
          ipMask=255.255.255.255
          ipOperation=is
     Forward Action
```

PPP Policy Group

Figure 33 shows a summary of the access policy for PPP subscribers.

Figure 33: PPP Policy Group



Name PPP

Description default policies for generic PPP users (same as DHCP)

Caption

Keywords

PL	DIR	PR	PRI	STA	CONDITIONS						ACTIONS
					CLA	SRC	DST	SVC	TOS	ARB	ACT
in	input	cp-to-ssp	500	□	content-provider-network-1	any	10.10.40.0/0.0.0.255	any	any	any	virtual_ipA...
					content-provider-network-2	any	172.16.0.0/0.0.255.255	any	any	any	
					CLA	SRC	DST	SVC	TOS	ARB	ACT
in	input	forward-to-SAE	200	□	sae	any	virtual_ipAddress/0.0.0.0	any	any	any	Forward
					CLA	SRC	DST	SVC	TOS	ARB	ACT
out	output	forward	500	□	any	any	any	any	any	any	Forward

The following information shows the configuration details of the PPP policy group in Figure 33.

policyGroupName=PPP, ou=junose, ou=sample, o=Policies, o=umc

PolicyList out

```
        name=out
        policyRoles=JUNOSE
        applicability=output

   PolicyRule: name=forward
        priority=500
        type=JUNOSE
        accountingRule=false
     Forward Action
```

PolicyList in

```
        name=in
        policyRoles=JUNOSE
        applicability=input

   PolicyRule: name=cp-to-ssp
        priority=500
        type=JUNOSE
        accountingRule=false
     ClassifyTrafficCondition content-provider-network-1
        DestinationNetwork:
          ipAddress=10.10.40.0
          ipMask=255.255.255.0
          ipOperation=is
     ClassifyTrafficCondition content-provider-network-2
        DestinationNetwork:
          ipAddress=172.16.0.0
          ipMask=255.255.0.0
          ipOperation=is
     NextHop Action
        nextHopAddress=virtual_ipAddress

   PolicyRule: name=forward-to-ssp
        priority=200
        type=JUNOSE
        accountingRule=false
     ClassifyTrafficCondition sae
        DestinationNetwork:
          ipAddress=virtual_ipAddress
          ipMask=255.255.255.255
          ipOperation=is
     Forward Action
```

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 1Mbps.
- 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.

Types of Policies

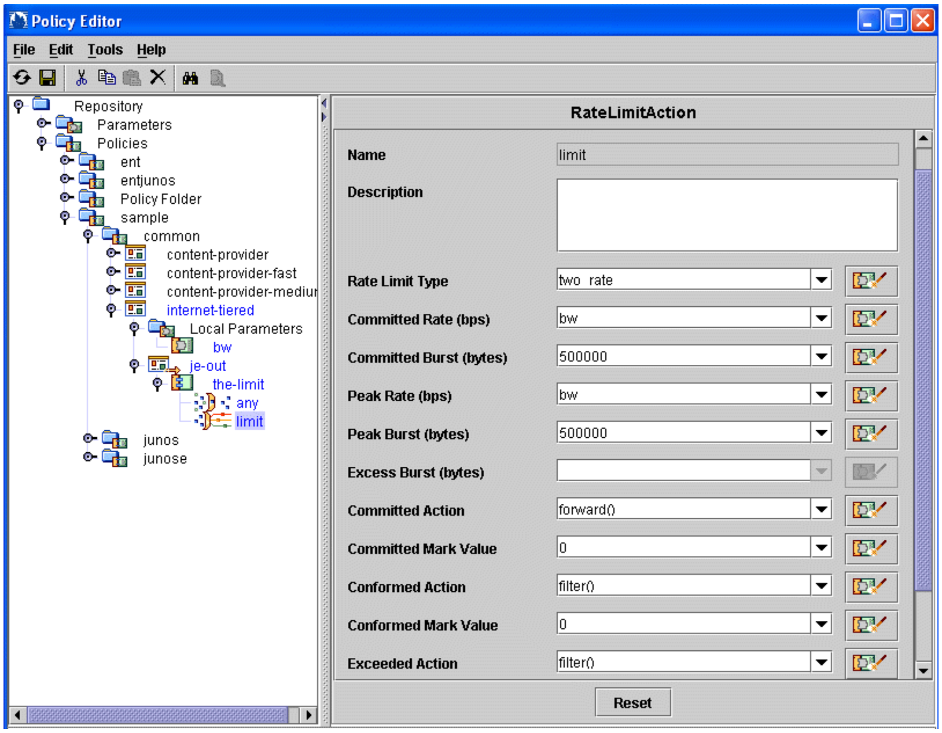
JUNOSe policies use the rate-limit action to control bandwidth, and JUNOS policies use the policer action to control bandwidth. You could also use QoS conditions and scheduler actions to provide tiered Internet services.

Sample JUNOSe Rate-Limiting Policy

The sample JUNOSe 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.

Figure 34: Rate-Limit Action for Tiered Internet Service



Name: limit

Description:

Rate Limit Type: two rate

Committed Rate (bps): bw

Committed Burst (bytes): 500000

Peak Rate (bps): bw

Peak Burst (bytes): 500000

Excess Burst (bytes):

Committed Action: forward()

Committed Mark Value: 0

Conformed Action: filter()

Conformed Mark Value: 0

Exceeded Action: filter()

The following information shows the configuration details of the Internet tiered policy group for JUNOSe routers.

policyGroupName=internet-tiered, ou=common, ou=sample, o=Policies, o=umc

Local Parameter

```
name=bw, defaultValue=5000000, parameterType=rate
```

PolicyList je-out

```
        name=je-out
        policyRoles=JUNOSE
        applicability=output

PolicyRule: name=the-limit
        priority=600
        type=JUNOSE
        accountingRule=true
     ClassifyTrafficCondition
     RateLimit Action
        rateLimitType=two_rate
        committedRate=bw
        committedBurst=500000
        peakRate=bw
        peakBurst=500000
        committed=Forward
        conformed=Filter
        exceeded=Filter
```

PolicyList je-in

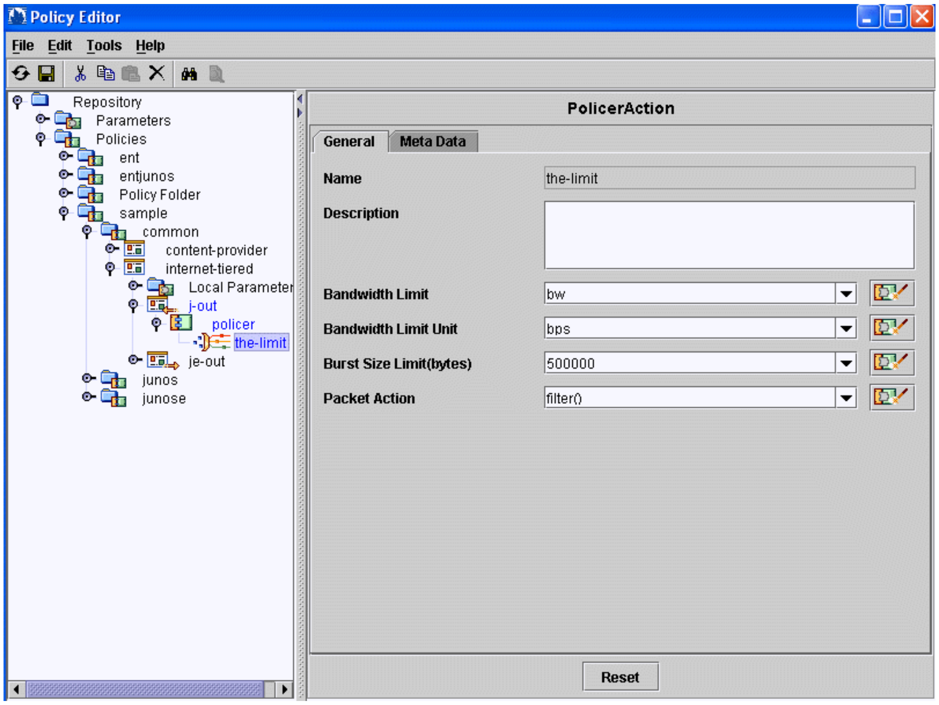
```
        name=je-in
        policyRoles=JUNOSE
        applicability=input

   PolicyRule: name=the-limit
        Priority=600
        type=JUNOSE
        accountingRule=true
     ClassifyTrafficCondition
     RateLimit Action
        rateLimitType=two_rate
        committedRate=bw
        committedBurst=500000
        peakRate=bw
        peakBurst=500000
        committed=Forward
        conformed=Filter
        exceeded=Filter
```

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.

Figure 35: Policer Action for Tiered Internet Service



Name: the-limit
Description:
Bandwidth Limit: bw
Bandwidth Limit Unit: bps
Burst Size Limit(bytes): 500000
Packet Action: filter()

The following information shows the configuration details of the Internet tiered policy group for JUNOS routing platforms.

policyGroupName=internet-tiered,ou=common,ou=sample,o=Policies,o=umc

Local Parameter

```
name=bw, defaultValue=5000000, parameterType=rate
```

PolicyList j-out

```
        name=j-out
        policyRoles=JUNOS
        applicability=output

PolicyRule: name=the-limit
        priority=600
        type=JUNOS
        accountingRule=true
     Policer Action
        bandwidthLimit=bw
        Burst=500000
        packetAction=filter
```

PolicyList j-in

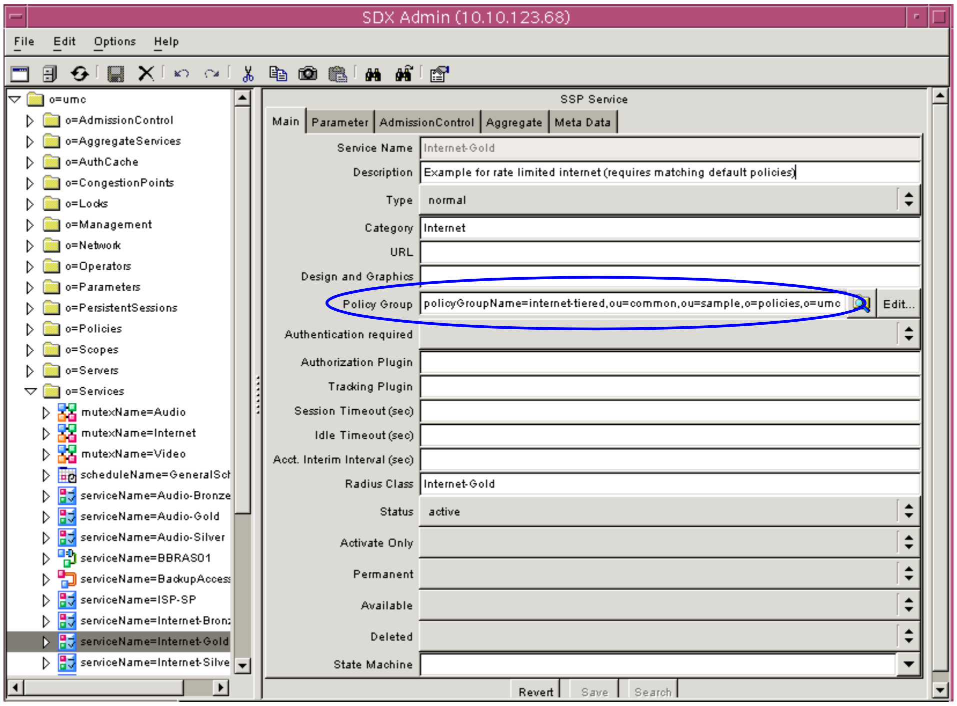
```
        name=j-in
        policyRoles=JUNOS
        applicability=input

     Policer Action
        bandwidthLimit=bw
        burst=500000
        packetAction=filter
```

Defining the Tiered Internet Services

You need to create three value-added (SSP) services—Gold, Silver, and Bronze.

As shown in Figure 36, assign to the new service one of the Internet-tiered policy groups that we created in the last section.

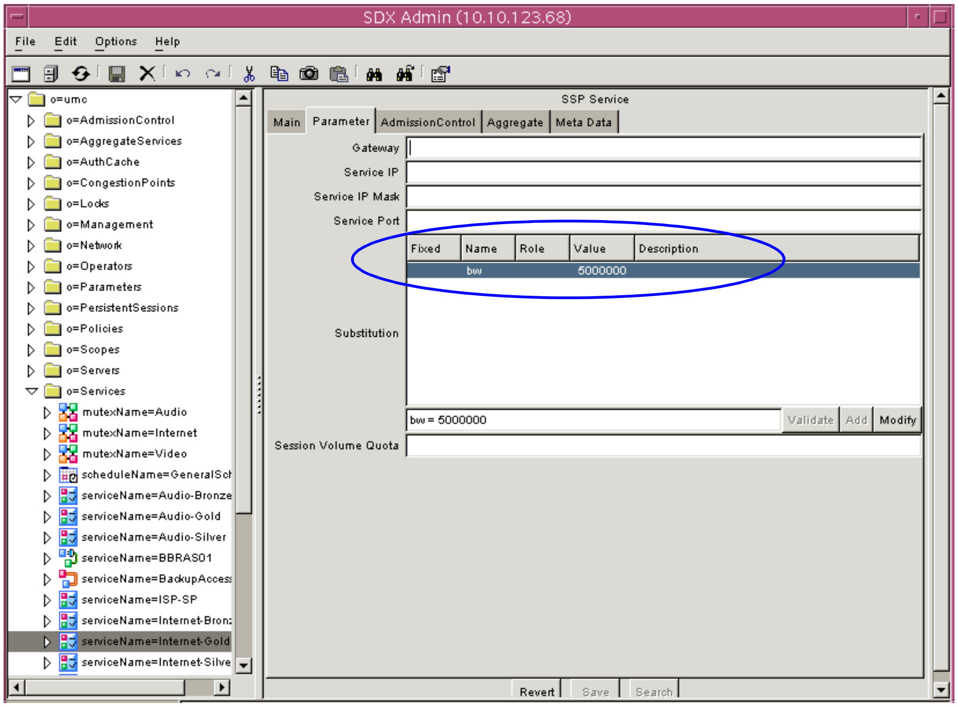
Figure 36: Sample Value-Added Service for Internet Gold Service



Service Name: Internet-Gold
Description: Example for rate limited internet (requires matching default policies)
Type: normal
Category: Internet
URL:
Design and Graphics:
Policy Group: policyGroupName=internet-tiered,ou=common,ou=sample,o=policies,o=umc
Authentication required:
Authorization Plugin:
Tracking Plugin:
Session Timeout (sec):
Idle Timeout (sec):
Acct. Interim Interval (sec):
Radius Class: Internet-Gold
Status: active
Activate Only:
Permanent:
Available:
Deleted:
State Machine:

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 1Mbps; and for the Bronze service, the bw value is 64 Kbps. Figure 37 shows how the substitution value is configured for the Gold service.

Figure 37: Parameter Pane of Internet Gold Service



Fixed	Name	Role	Value	Description
	bw		5000000	

bw = 5000000

Internet-Gold Service

```
serviceName=Internet-Gold,o=Services,o=umc
policyGroupName:internet-tiered,ou=common,ou=sample,o=Policies,o=umc
substitution:
  bw=5000000
```

Internet-Silver Service

```
serviceName=Internet-Silver,o=Services,o=umc
policyGroupName:internet-tiered,ou=common,ou=sample,o=Policies,o=umc
substitution:
  bw=1000000
```

Internet-Bronze Service

```
serviceName=Internet-Bronze,o=Services,o=umc
policyGroupName:internet-tiered,ou=common,ou=sample,o=Policies,o=umc
substitution:
  bw=64000
```

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.

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.

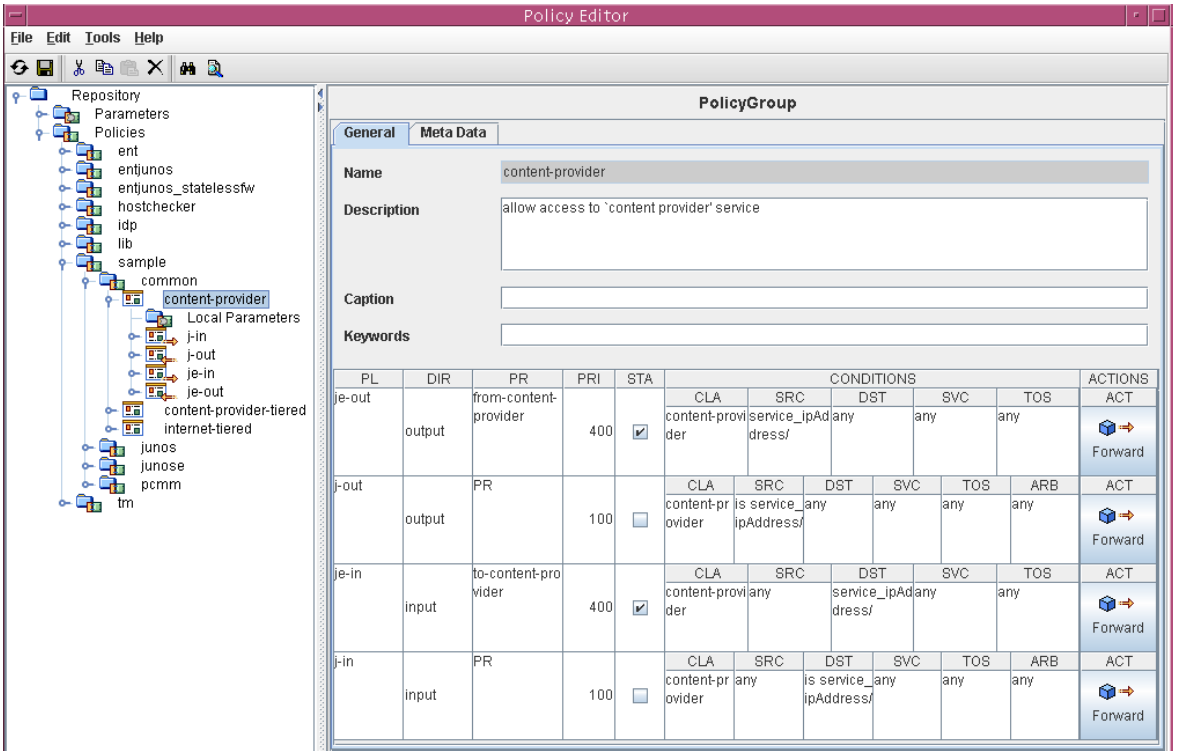
The policy rules in premium services typically have a higher priority (smaller precedence number) than other services and default policies. In this case, the policy rules in the content provider service policies have a priority of 400. The default policy rule has a priority of 500.

The default policy uses the next-hop action to send all traffic destined for the networks of these content providers to the SSP (see *Sample Access Policies* on page 380). 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. Figure 38 shows a summary of the content provider policy group.

Figure 38: Premium Service Policy Group



Name: content-provider

Description: allow access to `content provider' service

Caption:

Keywords:

PL	DIR	PR	PRI	STA	CONDITIONS						ACTIONS
					CLA	SRC	DST	SVC	TOS	ARB	ACT
je-out	output	from-content-provider	400	☑	content-provider	service_ipAddress/	any	any	any		Forward
j-out	output	PR	100	☐	content-provider	is service_ipAddress/	any	any	any	any	Forward
je-in	input	to-content-provider	400	☑	content-provider	any	service_ipAddress/	any	any		Forward
j-in	input	PR	100	☐	content-provider	any	is service_ipAddress/	any	any	any	Forward

The following information shows the configuration details of the premium service policy group shown in Figure 38.

policyGroupName=content-provider,ou=common,ou=sample,o=Policies,o=umc

PolicyList je-out

```
     name=je-out
     policyRoles=JUNOSE
     applicability=output

PolicyRule PR
     priority=400
     type=JUNOSE
     accountingRule=true

  ClassifyTrafficCondition
     SourceNetwork:
       ipAddress=service_ipAddress
       ipMask=service_ipMask
       ipOperation=is

  Forward Action
```

PolicyList j-out

```
        name=j-out
        policyRoles=JUNOS
        applicability=output

   PolicyRule PR
        priority=400
        type=JUNOS FILTER
        accountingRule=true

     ClassifyTrafficCondition
        SourceNetwork:
          ipAddress=service_ipAddress
          ipMask=service_ipMask
          ipOperation=is

     Forward Action
```

PolicyList je-in

```
        name=je-in
        policyRoles=JUNOSE
        applicability=input
   PolicyRule: name=PR
        priority=400
        type=JUNOSE
        accountingRule=true

     ClassifyTrafficCondition
        DestinationNetwork:
          ipAddress=service_ipAddress
          ipMask=service_ipMask
          ipOperation=is

     Forward Action
```

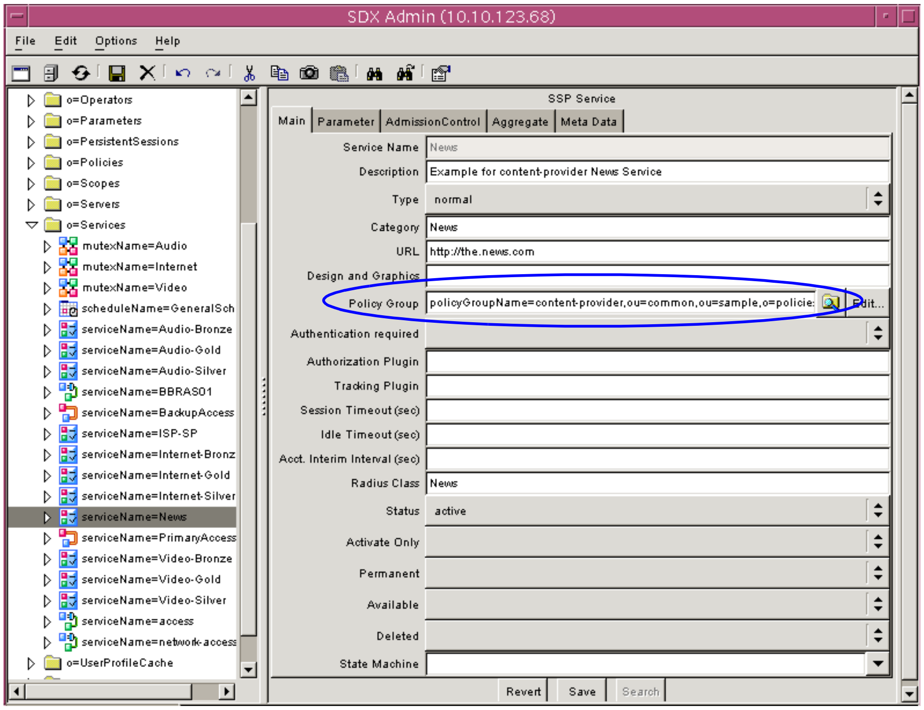
PolicyList j-in

```
        name=j-in
        policyRoles=JUNOS
        applicability=input
   PolicyRule: name=PR
        priority=400
        type=JUNOS FILTER
        accountingRule=true

     ClassifyTrafficCondition
        DestinationNetwork:
          ipAddress=service_ipAddress
          ipMask=service_ipMask
          ipOperation=is

     Forward Action
```

Defining the Premium Services

You need to create two value-added (SSP) services—one for the news service and one for the music service. As shown in Figure 39, assign to the new service- the content-provider policy group that we created in the last section.

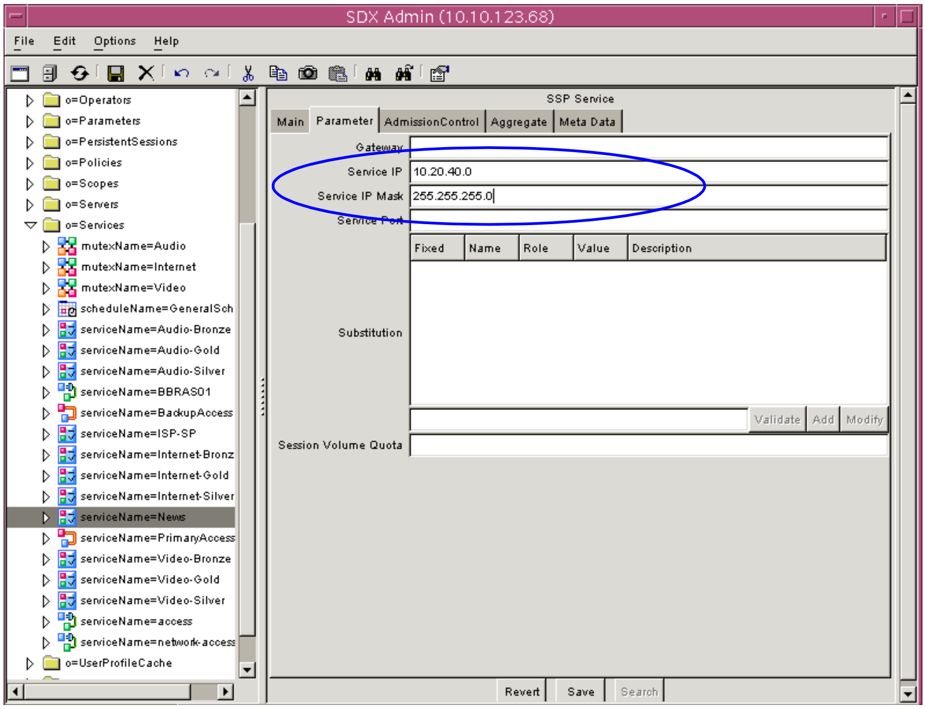
Figure 39: Sample Value-Added News Service



Service Name: News
Description: Example for content-provider News Service
Type: normal
Category: News
URL: http://the.news.com
Design and Graphics:
Policy Group: policyGroupName=content-provider,ou=common,ou=sample,o=policie:
Authentication required:
Authorization Plugin:
Tracking Plugin:
Session Timeout (sec):
Idle Timeout (sec):
Acct. Interim Interval (sec):
Radius Class: News
Status: active
Activate Only:
Permanent:
Available:
Deleted:
State Machine:

For each service, define a substitution value for the service_ipAddress and service_ipMask parameters. (See Figure 40.) Note that each content provider has a different service_ipAddress parameter.

Figure 40: Parameter Pane of News Service



Gateway

Service IP: 10.20.40.0

Service IP Mask: 255.255.255.0

Service Port

Substitution

Session Volume Quota

Music Service

The music service is provided by the XYZ company, which is a content provider.

```
serviceName=Music,o=Services,o=umc
policyGroupName: content-provider,ou=common,ou=sample,o=Policies,o=umc
substitution:
  service_ipAddress=10.20.30.0
  service_ipMask=255.255.255.0
```

News Service

The news service is provided by the ABC company, which is a content provider.

```
serviceName=News,o=Services,o=umc
policyGroupName: content-provider,ou=common,ou=sample,o=Policies,o=umc
substitution:
  service_ipAddress=10.20.40.0
  service_ipMask=255.255.255.0
```

Part 3
Generating Policies by Specifying Parameters

Chapter 14

Defining and Acquiring Values for Parameters

This chapter provides information about how the SAE acquires values for policies. Topics include:

Parameters and Substitutions

Each subscriber who uses the SRC network must appear 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. See *Chapter 8, Overview of Using Local and Global Parameters*.

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 (see *Example: Parameter Value Substitution* on page 411).

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 value-added 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:

   - *Restricting and Customizing Services for Subscribers* on page 27 in *Chapter 1, Managing Services with the SRC CLI*.

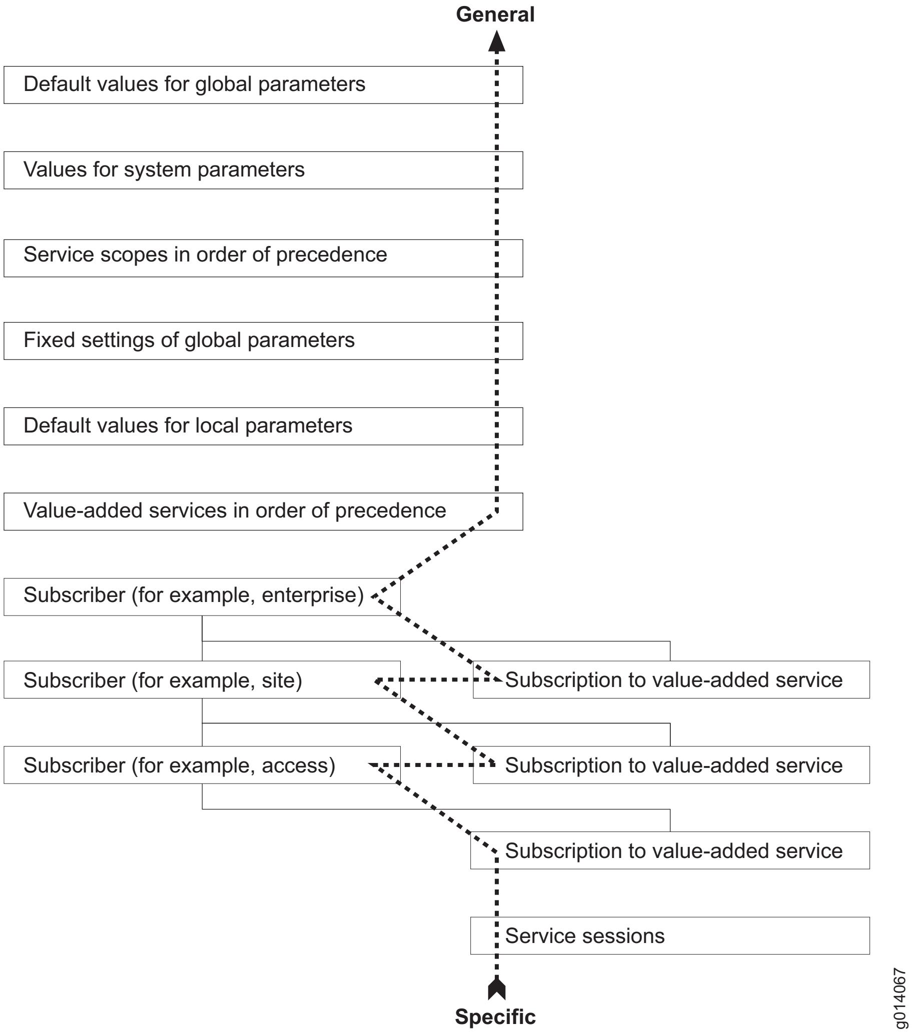
   - *Restricting and Customizing Services for Subscribers* on page 82 in *Chapter 2, Managing Services on a Solaris Platform*.

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 41 illustrates how the SAE selects values for a subscriber with one subscription to a service.

Figure 41: Value Acquisition for Single Subscriptions


General
Default values for global parameters
Values for system parameters
Service scopes in order of precedence
Fixed settings of global parameters
Default values for local parameters
Value-added services in order of precedence
Subscriber (for example, enterprise)
Subscription to value-added service
Subscriber (for example, site)
Subscription to value-added service
Subscriber (for example, access)
Subscription to value-added service
Service sessions
Specific
g014067

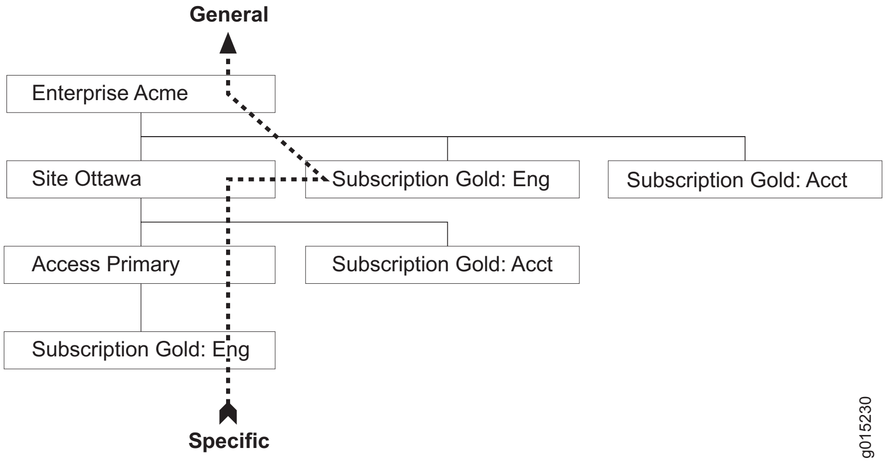
Value Acquisition for Multiple Subscriptions

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 42 shows an example that illustrates this concept.

Figure 42: 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 (see *Value Acquisition for Single Subscriptions* on page 398).

Defining Parameters

You can define parameters for different items in the SRC software. Depending on the item, you can define parameters with the SRC CLI, with LDAP clients, with SRC applications, or with other applications through SRC APIs.

Table 26 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 26: Parameter Definitions

Items That Contain Parameter Definitions	Methods by Which You Can Define the Parameter	Documentation That Describes How to Define Parameters
Global parameters, which you define in the runtime parameters in the policy repository folder.	SRC CLI Policy Editor SDX Admin	*Chapter 8, Overview of Using Local and Global Parameters*
Local parameters, which you define in policy groups.	SRC CLI Policy Editor	*Chapter 8, Overview of Using Local and Global Parameters*
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	Table 24 on page 196
Value-added services in order of the precedence associated with the scopes associated with the service	LDAP client SDX Admin	*Setting Parameters for Value-Added Services* on page 53 *Restricting and Customizing Services for Subscribers* on page 82
Services in order of the precedence associated with the scopes associated with the service	SRC CLI	*Setting Parameter Values for Services with the SRC CLI* on page 8 *Restricting and Customizing Services for Subscribers* on page 27
Subscribers	SRC CLI LDAP client SDX Admin	*SRC-PE Subscribers and Subscriptions Guide, Chapter 12, Configuring Subscribers and Subscriptions with the SRC CLI* *SRC-PE Subscribers and Subscriptions Guide, Chapter 13, Configuring Subscribers and Subscriptions with SDX Admin*

Table 26: Parameter Definitions (continued)

Items That Contain Parameter Definitions	Methods by Which You Can Define the Parameter	Documentation That Describes How to Define Parameters
Subscriptions	SRC CLI SDX Admin Residential portal or enterprise service portal Dynamic Service Activator SAE’s CORBA remote API	*SRC-PE Subscribers and Subscriptions Guide, Chapter 12, Configuring Subscribers and Subscriptions with the SRC CLI* *SRC-PE Subscribers and Subscriptions Guide, Chapter 13, Configuring Subscribers and Subscriptions with SDX Admin* *SRC-PE Sample Applications Guide* *SRC-PE Subscribers and Subscriptions Guide, Chapter 20, Overview of Enterprise Service Portals* *SRC Application Library Guide, Chapter 3, Activating Services with SOAP* SRC software distribution in the folder *SDK/doc/sae* or in the SAE CORBA remote API documentation on the Juniper Networks Web site at http://www.juniper.net/techpubs/software/management/sdx/api-index.html
Sessions	Residential portal Dynamic Service Activator SAE’s CORBA remote interface	*SRC-PE Sample Applications Guide* *SRC Application Library Guide, Chapter 3, Activating Services with SOAP* SRC software distribution in the folder *SDK/doc/sae* or in the SAE CORBA remote API documentation on the Juniper Networks Web site at http://www.juniper.net/techpubs/software/management/sdx/api-index.html

Formatting Substitutions

Some SRC components handle the substitution syntax for you. For example, Policy Editor allows 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 must enter some substitutions in SDX Admin using the correct syntax, however. Similarly, if you develop a portal that uses substitutions, you must 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 (see *Parameter Names* on page 403). 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 of the parameter (see *Roles* on page 403)
- <expression>—A definition for the parameter (see *Expressions* on page 404)
- //<comment>—A comment about a substitution that appears on a new line after the substitution syntax (see *Adding Comments to Substitutions* on page 410)

Parameter Names

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. For more information about parameter names, see *Specifying Parameter Names* on page 405.

Roles

Parameters fall into different categories, known as roles in SDX Admin and types in the Policy Editor. For example, a parameter that defines an IP address has the role address. For more information about roles, see Table 26 on page 401.

Expressions

An expression in a parameter definition can take one 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 ? rate : 2*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 *Using Operators* on page 407*)*.
- 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 the *Using Operators* on page 407.

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 27 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 27: 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
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
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

Table 27: Operations That You Can Use in Expressions (continued)

Operation	Operator	Number of Arguments	Result If Different from Java Conventions	Conditions for Evaluation	Example
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)
Minimum of the arguments, min() = +infinity	min()	Zero or more		All arguments must be numbers	min (1, 3, 2, 4)
Negation	–	One		Argument must be a number	–5
Not equal	!=	Two	Nonzero integer if the arguments are not equal	Both arguments must be numbers	rate !=5
Not match	not	One		None – expressions with this operator cannot be evaluated	not 192.0.2.1
Product of the arguments	*	Two		Both arguments must be numbers	rate*2
Raise the first argument to the power of the second argument	**	Two		Both arguments must be numbers	2**16
Range from the first argument to the second argument	..	Two		None—expressions with this operator cannot be evaluated	0..49
Remainder of division of the first argument by the second argument	%	Two		Both arguments must be integers	5%2
Round off the argument to the closest number	round()	One	Integer closest to the argument	Argument must be numbers	round(986532.654)
Round the argument down	floor()	One	Biggest integer less than or equal to the argument	Argument must be numbers	floor (986532.654)
Round the argument up	ceiling()	One	Smallest integer greater than or equal to the argument	Argument must be numbers	ceiling (986532.654)

Table 27: Operations That You Can Use in Expressions (continued)

Operation	Operator	Number of Arguments	Result If Different from Java Conventions	Conditions for Evaluation	Example
Shift the first argument left by the number of bits in the second argument	< <	Two		Both arguments must be integers	986532 < < 2
Shift the first argument right by the number of bits in the second argument	> >	Two		Both arguments must be integers	986532 > >2
Sum of the arguments	+	One or two		Both arguments must be numbers	876 + 345 +855

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)
- ~,!

Adding Comments to Substitutions

You can add a comment on the last line of the substitution. To do so:

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.

Validating Substitutions

You can validate substitutions with Policy Editor, SDX Admin, and the Enterprise portal. For example, if you enter a substitution for a value-added service with SDX Admin, you can validate that substitution with SDX Admin.

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.

Example: Parameter Value Substitution

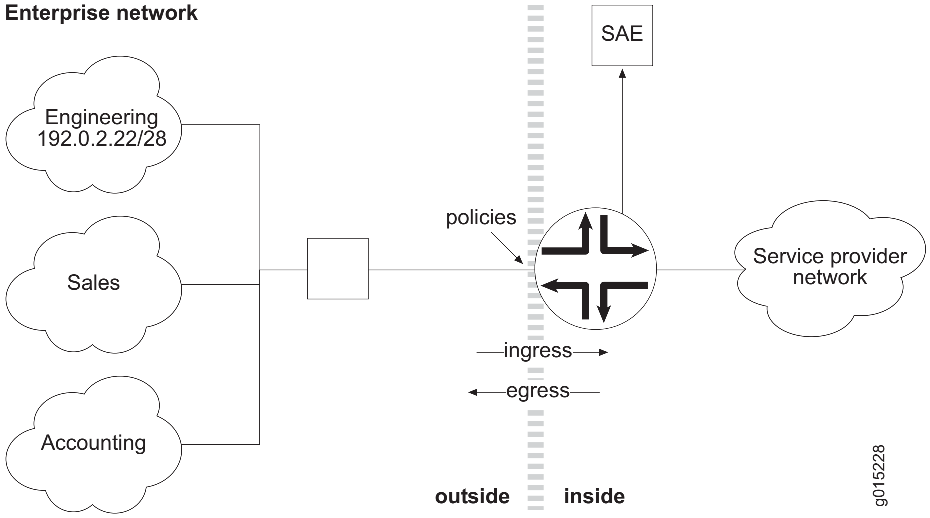
This section provides an example of how to use parameters and substitutions. It contains the following sections:

- Setting Up a Service That Uses Parameters on page 411
- Acquiring the Parameter Values on page 423

Setting Up a Service That Uses Parameters

In this example, we will create a value-added service that provides a gold-level quality of service. We will then subscribe this service to a department subnet in an enterprise network and be able to track and charge the department for the volume of bandwidth used. Figure 43 shows the network in our example.

Figure 43: 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.

The example uses two types of parameters (note that SDX Admin uses the term role in place of type):

- rate—Used to scale the rate limiter
- network—Used to specify IP subnets in classify conditions

Summary of Procedure

The following is a summary of the procedure we will use to set up the example.

1. Create 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
2. Create a value-added service called GoldMetered, and assign tierpolicy as the policy group. In the GoldMetered service, configure 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.
3. Create an enterprise subscriber, and configure 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.

4. Subscribe the subscriber to the GoldMetered service, and configure the following parameter substitution:

   - dept—Set to eng.

Creating a Policy Group

Use Policy Editor to create a policy group.

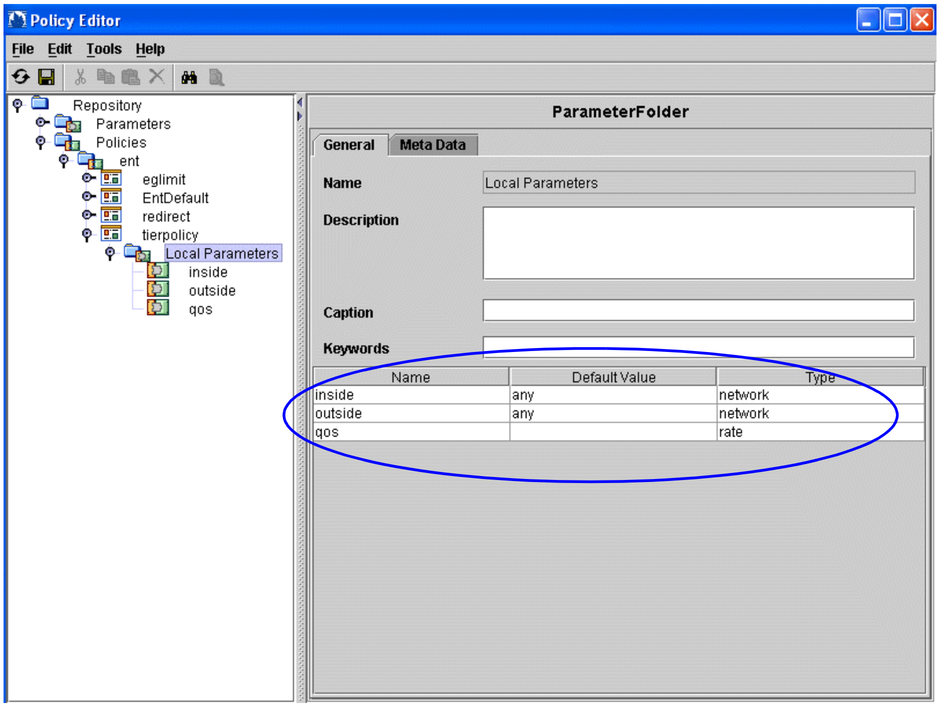
1. Create a policy group called tierpolicy.

2. Create the following local parameters, which are parameters that will be used only with tierpolicy.

   - 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

   - 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

   - qos—Rate parameter



Name: Local Parameters

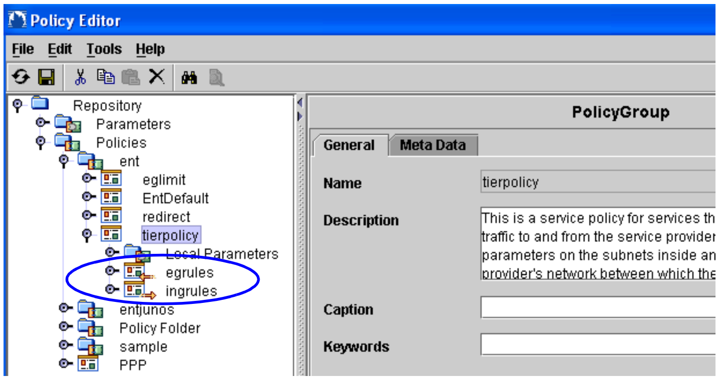
Description:

Caption:

Keywords:

Name	Default Value	Type
inside	any	network
outside	any	network
qos		rate

3. Create two policy lists, one for the ingress side of the interface, and one for the egress side of the interface.



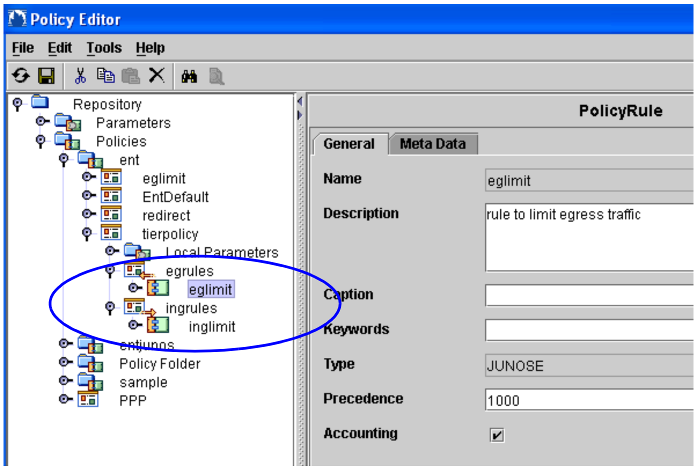
Name: tierpolicy

Description: This is a service policy for services th
traffic to and from the service provider
parameters on the subnets inside an
provider's network between which the

Caption:

Keywords:

4. Create two policy rules, one for ingress traffic and one for egress traffic.



Name: eglimit

Description: rule to limit egress traffic

Caption:

Keywords:

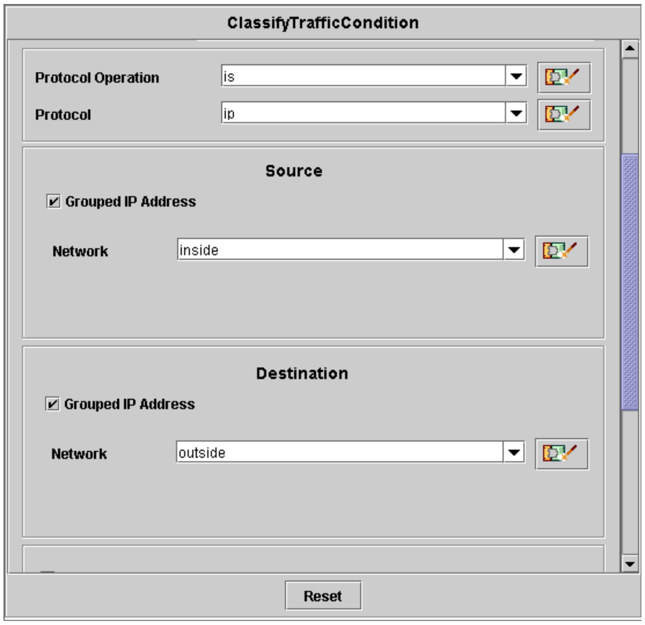
Type: JUNOSE

Precedence: 1000

Accounting: ☑

5. 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



Protocol Operation: is

Protocol: ip

Source

☑ Grouped IP Address

Network: inside

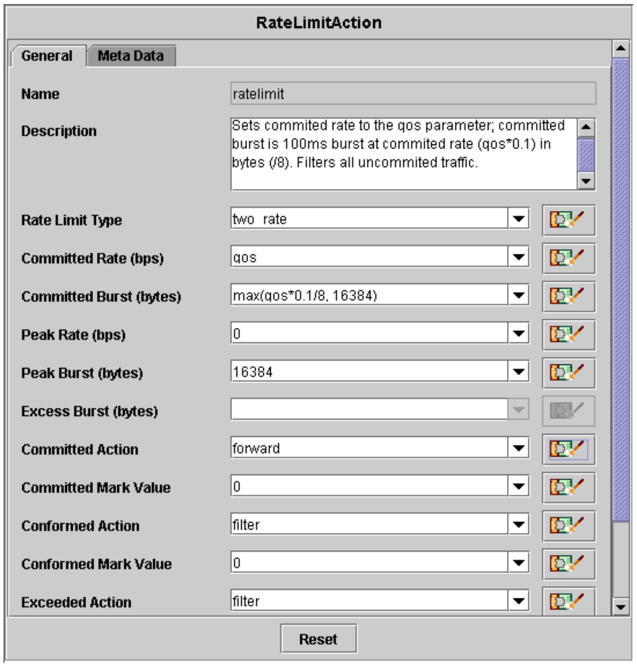
Destination

☑ Grouped IP Address

Network: outside

6. Also in the egress policy rule, create a rate-limit action that does the following:

    - Sets the committed rate to the qos parameter.
    - Sets the committed burst to the maximum of either 100 ms burst at committed rate (qos*0.1) in bytes (/8) or 16384.
    - Sets the peak burst to 16384.
    - Forwards all committed traffic.
    - Filters all uncommitted traffic.



Name: ratelimit

Description: Sets commited rate to the qos parameter; committed burst is 100ms burst at commited rate (qos*0.1) in bytes (/8). Filters all uncommited traffic.

Rate Limit Type: two_rate

Committed Rate (bps): qos

Committed Burst (bytes): max(qos*0.1/8, 16384)

Peak Rate (bps): 0

Peak Burst (bytes): 16384

Excess Burst (bytes):

Committed Action: forward

Committed Mark Value: 0

Conformed Action: filter

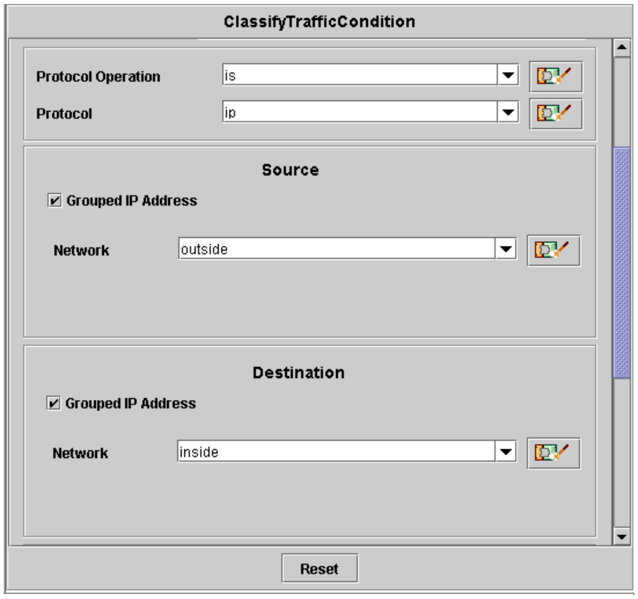
Conformed Mark Value: 0

Exceeded Action: filter

7. 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



Protocol Operation: is

Protocol: ip

Source

☑ Grouped IP Address

Network: outside

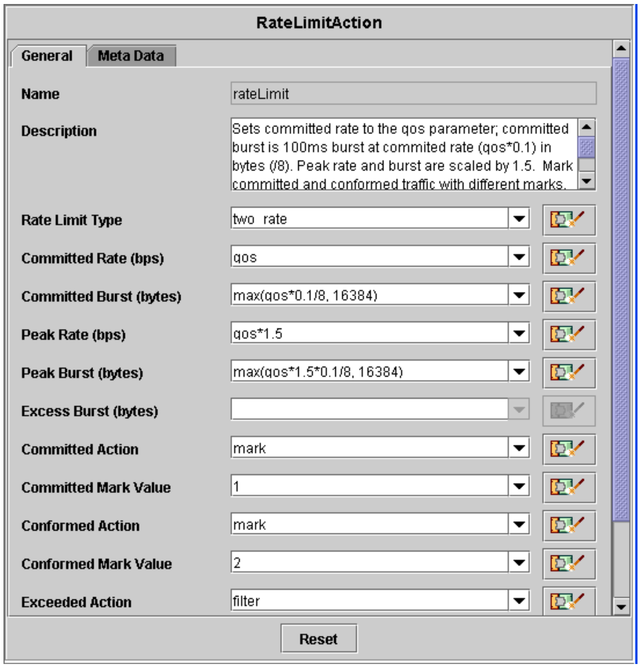
Destination

☑ Grouped IP Address

Network: inside

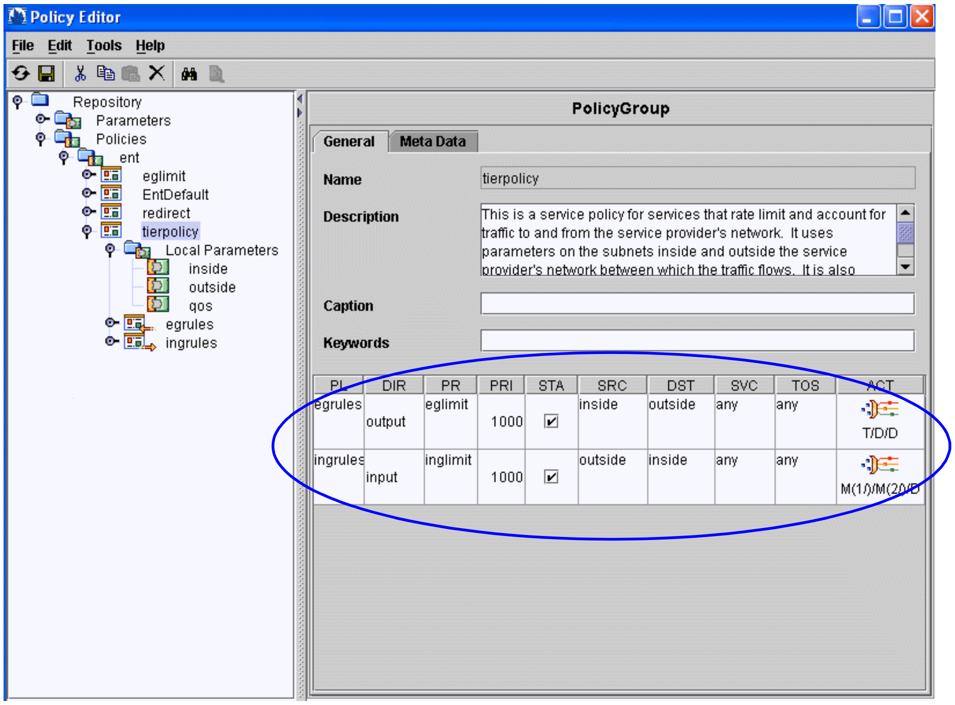
8. Also in the ingress policy rule, create a rate-limit action that does the following:

   - Sets the committed rate to the qos local parameter.
   - Sets the committed burst to the maximum of either 100 ms burst at the committed rate (qos*0.1) in bytes (/8) or 16384.
   - Scales the peak rate and burst by 1.5.
   - Marks committed and conformed traffic with different marks (1 and 2).
   - Drops all traffic that exceeds the rate limit.



Name	rateLimit
Description	Sets committed rate to the qos parameter; committed burst is 100ms burst at commited rate (qos*0.1) in bytes (/8). Peak rate and burst are scaled by 1.5. Mark committed and conformed traffic with different marks.
Rate Limit Type	two_rate
Committed Rate (bps)	qos
Committed Burst (bytes)	max(qos*0.1/8, 16384)
Peak Rate (bps)	qos*1.5
Peak Burst (bytes)	max(qos*1.5*0.1/8, 16384)
Excess Burst (bytes)	
Committed Action	mark
Committed Mark Value	1
Conformed Action	mark
Conformed Mark Value	2
Exceeded Action	filter

The policy group should now look like this:



Name: tierpolicy

Description: This is a service policy for services that rate limit and account for traffic to and from the service provider's network. It uses parameters on the subnets inside and outside the service provider's network between which the traffic flows. It is also

Caption:

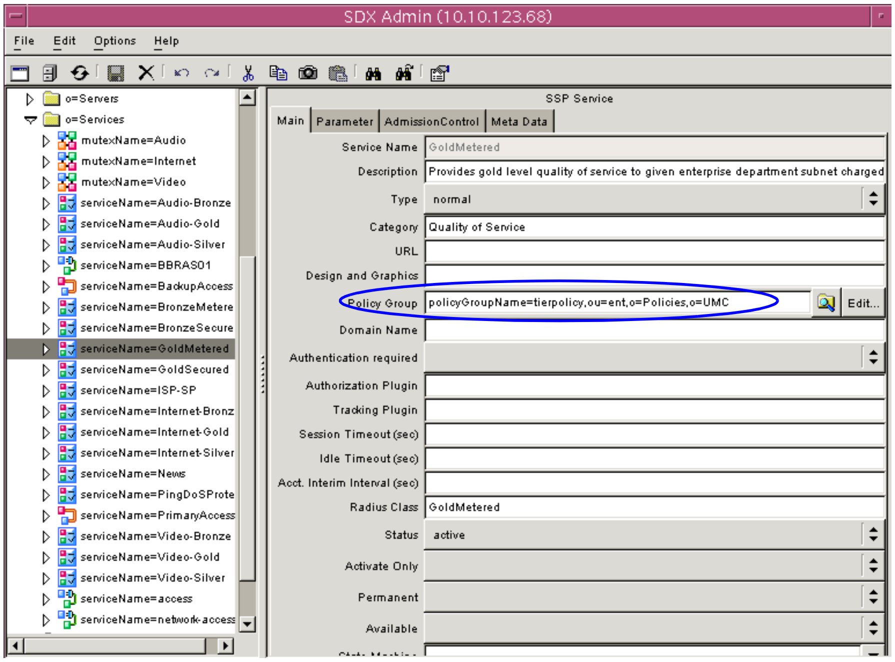
Keywords:

PL	DIR	PR	PRI	STA	SRC	DST	SVC	TOS	ACT
egrules	output	eglimit	1000	☑	inside	outside	any	any	T/D/D
ingrules	input	inglimit	1000	☑	outside	inside	any	any	M(1/)/M(2/)/D

Creating a Value-Added Service

Use SDX Admin to create a value-added service.

1. Create a value-added service called GoldMetered, and assign tierpolicy as the policy group.



- o=Servers
- o=Services
  - mutexName=Audio
  - mutexName=Internet
  - mutexName=Video
  - serviceName=Audio-Bronze
  - serviceName=Audio-Gold
  - serviceName=Audio-Silver
  - serviceName=BBRAS01
  - serviceName=BackupAccess
  - serviceName=BronzeMetere
  - serviceName=BronzeSecure
  - serviceName=GoldMetered
  - serviceName=GoldSecured
  - serviceName=ISP-SP
  - serviceName=Internet-Bronz
  - serviceName=Internet-Gold
  - serviceName=Internet-Silver
  - serviceName=News
  - serviceName=PingDoSProte
  - serviceName=PrimaryAccess
  - serviceName=Video-Bronze
  - serviceName=Video-Gold
  - serviceName=Video-Silver
  - serviceName=access
  - serviceName=network-access

Service Name	GoldMetered
Description	Provides gold level quality of service to given enterprise department subnet charged
Type	normal
Category	Quality of Service
URL	
Design and Graphics	
Policy Group	policyGroupName=tierpolicy,ou=ent,o=Policies,o=UMC
Domain Name	
Authentication required	
Authorization Plugin	
Tracking Plugin	
Session Timeout (sec)	
Idle Timeout (sec)	
Acct. Interim Interval (sec)	
Radius Class	GoldMetered
Status	active
Activate Only	
Permanent	
Available	

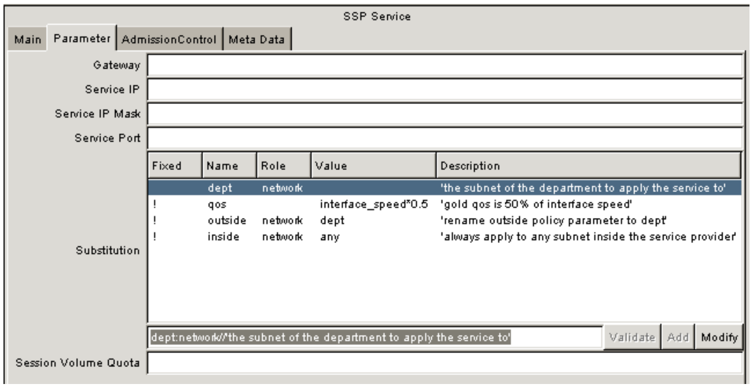
2. Select the **Parameter** tab of the GoldMetered service, and add the following parameters to the substitution table:

   - 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.



Fixed	Name	Role	Value	Description
	dept	network		'the subnet of the department to apply the service to'
!	qos		interface_speed*0.5	'gold qos is 50% of interface speed'
!	outside	network	dept	'rename outside policy parameter to dept'
!	inside	network	any	'always apply to any subnet inside the service provider'

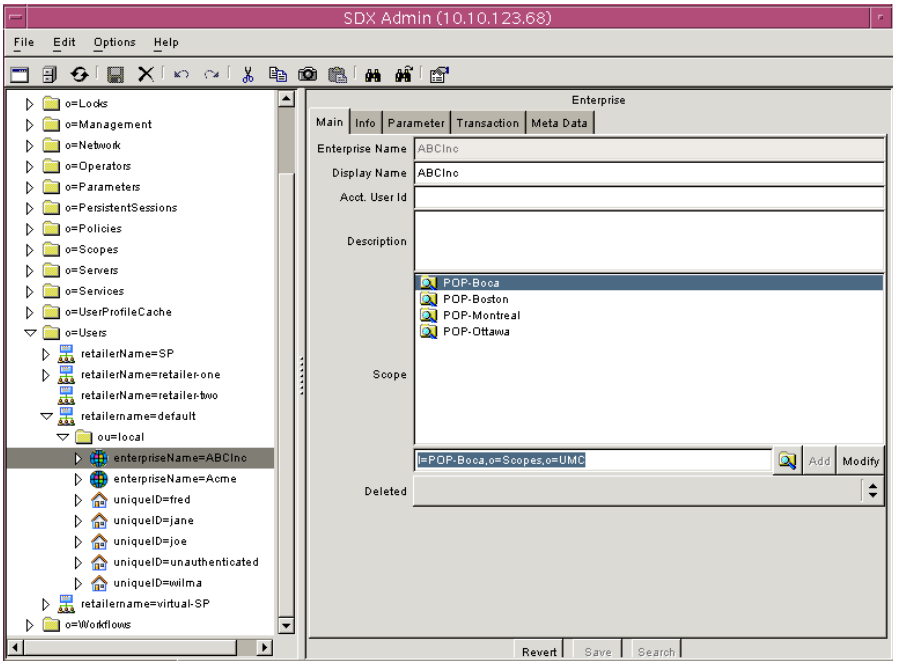
dept:network//'the subnet of the department to apply the service to'

Creating an Enterprise Subscriber

The next step is to create an enterprise subscriber. Within the subscriber definition, create a parameter called eng that is parameter type (role) network, and set the value of eng to 192.0.2.22/28.

You create a subscriber by using SDX Admin or another directory client. You can create the eng parameter with SDX Admin or the sample enterprise service portal.

1. In SDX Admin, create an enterprise subscriber called ABCInc.



Enterprise Name: ABCInc

Display Name: ABCInc

Acct. User Id:

Description:

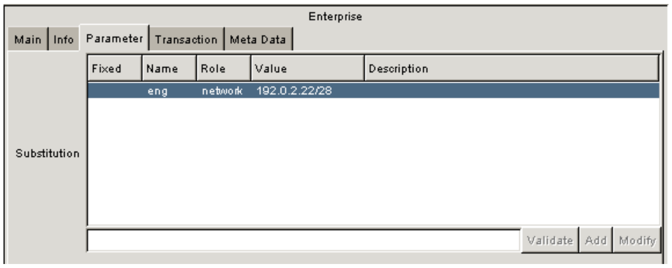
Scope: POP-Boca, POP-Boston, POP-Montreal, POP-Ottawa

l=POP-Boca,o=Scopes,o=UMC

Deleted:

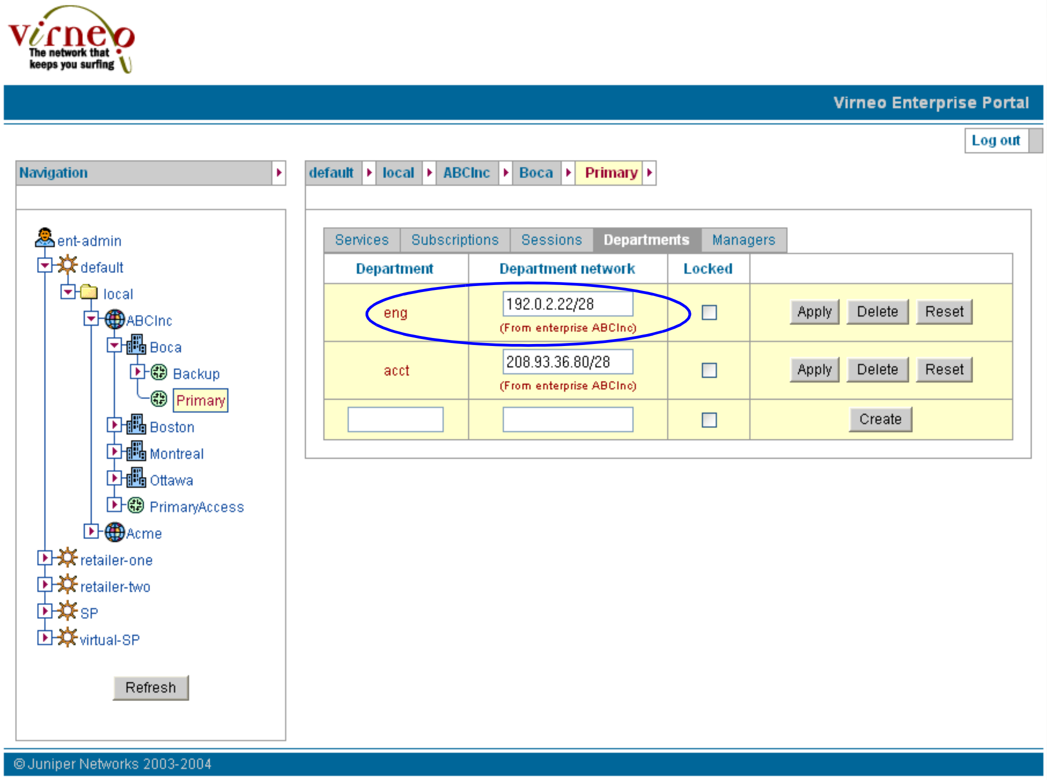
2. Create the eng parameter as part of the subscriber definition. You can perform this step by using either SDX Admin or the sample enterprise service portal.

   - To create the eng parameter in SDX Admin, select the **Parameter** tab of the ABCInc subscriber, and add the eng parameter to the substitution table.



Fixed	Name	Role	Value	Description
	eng	network	192.0.2.22/28	

   - 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.



Department	Department network	Locked	
eng	192.0.2.22/28 (From enterprise ABCInc)	□	Apply Delete Reset
acct	208.93.36.80/28 (From enterprise ABCInc)	□	Apply Delete Reset
		□	Create

Subscribing ABCInc to the GoldMetered Service

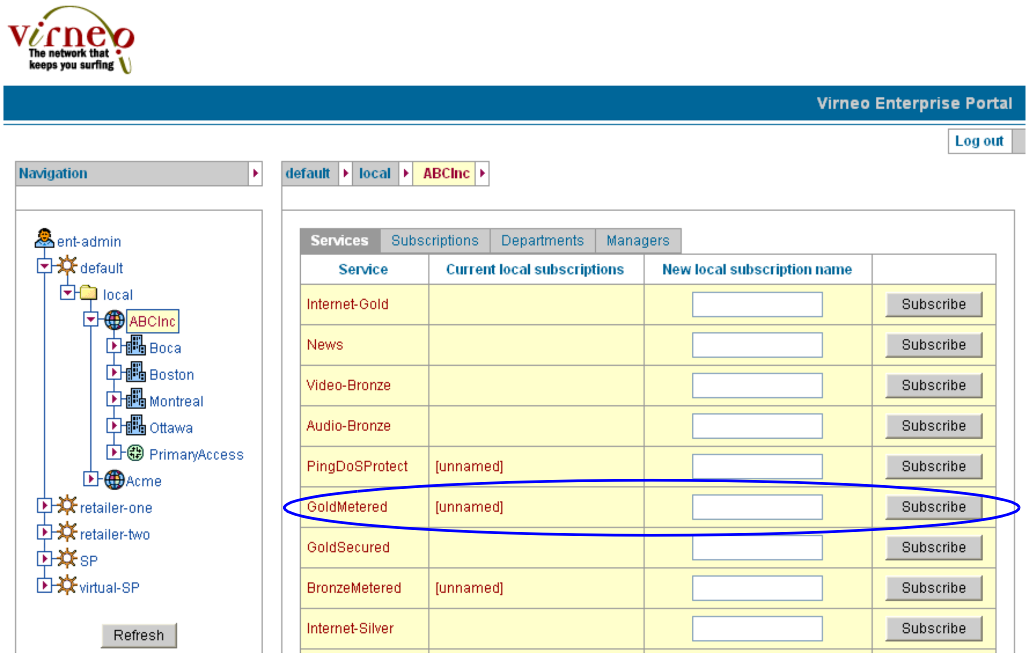
Next, subscribe the ABCInc subscriber to the GoldMetered service. You can perform this step by using SDX Admin or the sample enterprise service portal.

In the sample enterprise service portal:

1. Select **ABCInc.** in the navigation pane.

2. Select the **Services** tab.

   The Services pane appears.

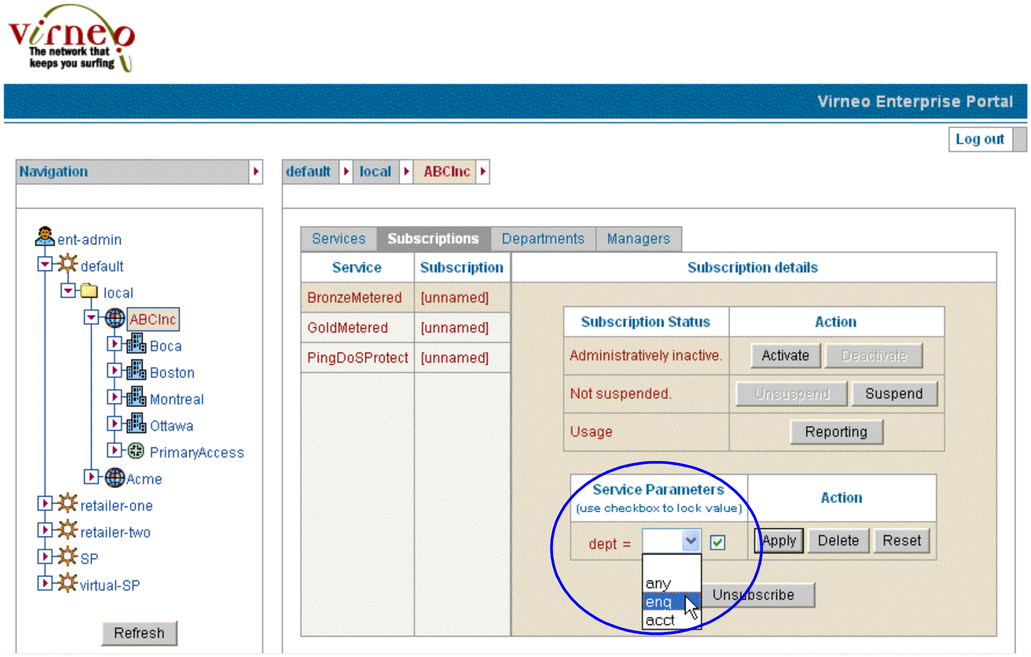


Service	Current local subscriptions	New local subscription name	
Internet-Gold			Subscribe
News			Subscribe
Video-Bronze			Subscribe
Audio-Bronze			Subscribe
PingDoSProtect	[unnamed]		Subscribe
GoldMetered	[unnamed]		Subscribe
GoldSecured			Subscribe
BronzeMetered	[unnamed]		Subscribe
Internet-Silver			Subscribe

3. Click **Subscribe** in the GoldMetered service row.

4. Select the **Subscriptions** tab.

   The Subscriptions pane appears.



Service	Subscription
BronzeMetered	[unnamed]
GoldMetered	[unnamed]
PingDoSProtect	[unnamed]

Subscription Status	Action
Administratively inactive.	Activate Deactivate
Not suspended.	Unsuspend Suspend
Usage	Reporting

Service Parameters (use checkbox to lock value)	Action
dept = ☑	Apply Delete Reset

5. In the dept = field of the Service Parameters box, set the value of the dept parameter to eng.

Acquiring the Parameter Values

Once 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:

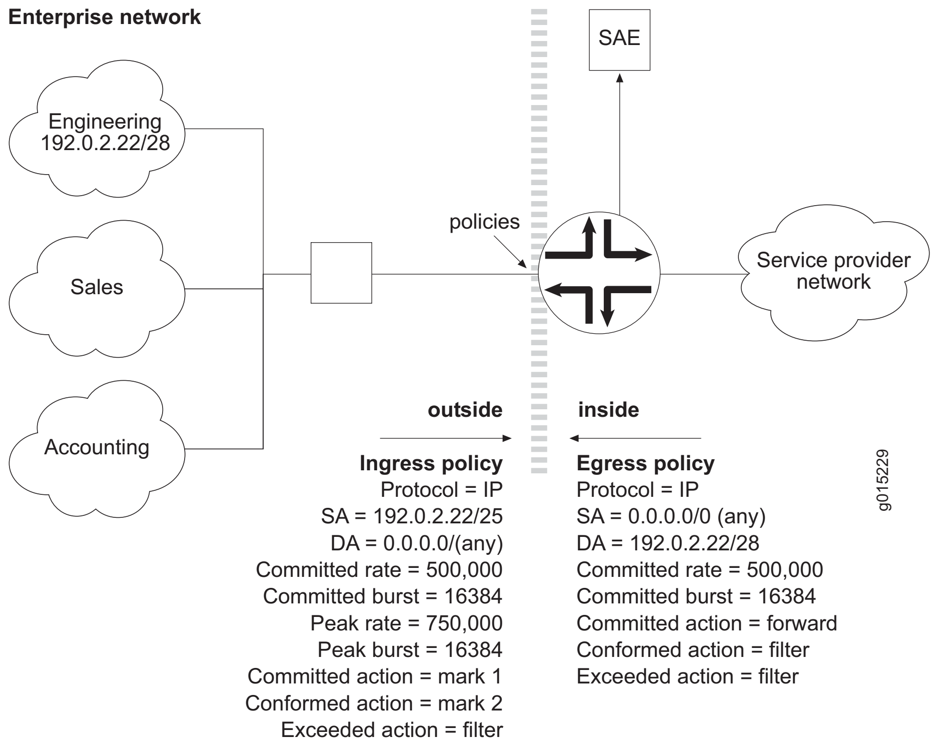
  - outside = dept—Acquired from the service definition
  - dept = eng—Acquired from the subscription
  - eng = 192.0.2.22/28—Acquired from the enterprise subscriber definition
  - qos = 500,000

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 44 shows the values of the ingress and egress policies that are applied to the router in our sample network.

Figure 44: Policies Applied to the Sample Network



Ingress policy
Protocol = IP
SA = 192.0.2.22/25
DA = 0.0.0.0/(any)
Committed rate = 500,000
Committed burst = 16384
Peak rate = 750,000
Peak burst = 16384
Committed action = mark 1
Conformed action = mark 2
Exceeded action = filter

Egress policy
Protocol = IP
SA = 0.0.0.0/0 (any)
DA = 192.0.2.22/28
Committed rate = 500,000
Committed burst = 16384
Committed action = forward
Conformed action = filter
Exceeded action = filter

Index